

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

**Zmapování výrobních procesů za pomoci nástroje Enterprise
Architect**

Diplomová práce

Autor: Bc. Tomáš Lepš
Studijní obor: Informační management – Navazující magisterské

Vedoucí práce: doc. Ing. Hana Tomášková, Ph. D.

Hradec Králové

Duben 2018

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 19.4.2018

Tomáš Lepš

Poděkování:

Na tomto místě bych rád poděkoval vedoucí práce doc. Ing. Haně Tomáškové, Ph.D. za její vedení a cenné rady při zpracování této práce. Dále bych rád vyjádřil poděkování vedení firmy za poskytnutí podkladů k zpracování této práce a získání cenných praktických zkušeností.

Anotace

Tato diplomová práce se zabývá představením a následným použitím modelovacích jazyků BPMN a ArchiMate, které slouží ke grafické vizualizaci podnikové architektury. Cílem této práce je popsání základních principů a použití elementů modelovacích jazyků. Praktická část se skládá z popisu, jak správně modelovat. Toto modelování je následně prezentováno na reálném příkladu při analýze firmy implementující nový systém. Zvyšování efektivity produkce v dnešní době není lehkým úkolem a je na každé firmě, jak se k tomuto problému postaví. Znalost své firmy a její zpracování do přehledných grafických modelů je nicméně prvním vhodným krokem, jak tuto efektivitu zvednout.

Hlavním přínosem této práce je nastínění a vysvětlení potřebných postupů pro čtenáře, kteří budou chtít modelovací notace BPMN a ArchiMate využívat. A ukázka aplikace těchto jazyků v praxi.

Klíčová slova:

Procesní mapování, výroba, BPMN, ArchiMate, Enterprise Architect

Annotation

Title: Charting manufacturing processes with the help of tools Enterprise Architect

This diploma thesis deals with the presentation and subsequent use of the BPMN and ArchiMate modelling languages, which are used for graphical visualization of corporate architecture. The aim of this work is to describe basic principles and use of modelling language elements. The practical part consists of a description of how to model correctly. This modelling is presented on a realistic example when analysing the company implementing a new system. Raising the effectivity of production today is not an easy task heavily depending on the companys' approach in dealing with the issue. Knowledge of your business and its processing into clear graphical models is nevertheless the first appropriate step to raise that efficiency.

The main benefit of this work is to outline and explain the necessary procedures for the readers who want to use the modelling notation BPMN and ArchiMate. And a demonstration of the application of these languages in practice.

Key words:

Process Mapping, Production, BPMN, ArchiMate, Enterprise Architect

Obsah

1	Úvod.....	1
2	Cíl práce.....	3
3	Teoretická část	4
3.1	Výroba.....	4
3.1.1	Charakteristika výrobních faktorů	4
3.1.2	Přístupy v řízení výroby.....	5
3.1.3	Industry 4.0.....	11
3.2	Procesní řízení.....	12
3.3	BPMN.....	14
3.3.1	Vývoj BPMN	15
3.3.2	Elementy syntaxe BPMN.....	17
3.4	ArchiMate	34
3.4.1	Motivation Elements (Motivační elementy).....	35
3.4.2	Application layer (Aplikační vrstva)	49
3.4.3	Technology Layer (Technologická vrstva)	55
3.4.4	Physical Elements (Fyzické elementy).....	62
3.4.5	Vazby (vztahy) mezi elementy	65
3.5	Enterprise Architect.....	75
4	Praktická část.....	77
4.1	Přestavení společnosti	78
4.2	Modelování procesů	78
4.3	Postup tvorby grafického modelování	80
4.3.1	Organizační struktura obchodního modelu	81
4.3.2	Tvorba byznys vrstvy.....	81
4.3.3	Tvorba cílů.....	85

4.3.4	Tvorba aplikační vrstvy.....	86
4.3.5	Tvorba technologické vrstvy.....	86
4.4	Výroba – grafické mapování.....	86
4.4.1	Organizační struktura obchodního modelu	87
4.4.2	Modelování byznys procesů ERP systému.....	88
4.4.3	Aplikační vrstva.....	97
4.5	Modelování u zákazníka	98
4.5.1	Definování cílů	98
4.5.2	Modelování současného stavu výrobní firmy.....	99
4.6	Navržení cílového řešení.....	109
4.6.1	Zhodnocení konečného stavu	116
5	Shrnutí výsledků.....	118
6	Závěry a doporučení	120
7	Seznam použité literatury.....	122

1 Úvod

V dnešní době se informační a komunikační technologie staly nedílnou součástí každého podnikání. Stěží by se ještě dal najít podnik, ve kterém by se nenacházel nějaký důležitý software a hardware, bez něhož by se fungování firmy značně ztížilo nebo dokonce úplně znemožnilo. Rozvíjení informačních technologií ve společnostech vedlo k pokroku. Firmy, které měly možnost využívat systémy, jež napomáhaly k automatizaci procesů, disponovaly na trhu značnou výhodou a získávaly náskok před konkurencí. V současnosti nejsou rozdíly v informačních technologiích tak znatelné, jak tomu bylo dříve. I přesto je na ně kladen velký důraz a vytvářet tlak za účelem maximální podpory byznysu a produkce nových obchodních příležitostí. Proto vznikly grafické notace, které mají odrážet klíčové oblasti podnikové architektury z pohledu byznysu a IT. Tyto metody je možné použít k identifikaci slabých míst podniku a navrhnout potřebná opatření pro jejich nápravu.

Grafická podniková architektura je momentálně jedním z nejdůležitějších nástrojů, který by měla firma řešit, aby zlepšila svůj aktuální stav na trhu. Ucelený pohled na podnik a jeho procesy umožní společnosti přinést mnohem hlubší pohled na specifický problém.

Grafická notace nabízí propracovanější pohled na celou podnikovou architekturu v porovnání s lety minulými, kdy byly hlavními pracovními pomůckami pero a papír.

Grafické mapování má ale i své nevýhody, a to především časovou náročnost modelování a nízkou dostupnost souhrnných materiálů, ze kterých se dá čerpat. Pokud se však tato překážka překoná, čeká firmu velký krok vřed. Druhým úskalím jsou osoby, které grafické mapování odmítají, protože ho vidí jako přítěž a nutnost učit se něčemu novému. Myslí si, že současný stav je dostačující a změny nejsou třeba. Grafické modely nejsou však nic jiného, než přepracovaná dokumentace, kterou firma již má.

Nejlepším způsobem, jak zákazníka do dané problematiky zasvětit, je mít zpracované vlastní procesy nebo celou podnikovou architekturu odrážející oblast zájmu mezi komunikujícími stranami. V případě firmy nabízející nějaký systém je

vhodné, aby měla tento systém zmapovaný do grafické podoby a mohla jej zákazníkovi prezentovat na příslušných diagramech. V dalším kroku je vhodné klientovi navrhnout vytvoření jeho podnikové architektury jako součást přípravy na vytváření a implementaci nového systému. K tomuto účelu je možné použít modelovací syntaxe pro popis podnikové architektury ArchiMate a BPMN, které se více zaměřují na procesní modelování.

2 Cíl práce

Tato práce si klade za cíl především zmapování výrobních procesů do grafické podoby. Je ale také nutné čtenářům představit a popsat použité syntaxe BPMN a ArchiMate. Bez toho by bylo velmi náročné pochopit jejich zapojení do praktických příkladů. Dalším záměrem je vytvoření ucelených postupů, podle kterých bude možné modelovat pohled na systém a také zbytek společnosti za pomoci propojení zmíněných dvou syntaxí.

Teoretická část

V teoretické části práce bude čtenář seznámen s výrobní oblastí, která bude graficky modelována v rámci modulu ERP systému. Výrobní část je popsána na takové úrovni, aby čtenář získal ucelenou informaci o dané oblasti a postupech v ní používaných. Dále bude objasněn rozdíl mezi funkčním a procesním přístupem. Procesní přístup je dnes jedním z nejmodernějších postupů řízení firmy. V druhé části budou popsány grafické notace, jejich význam a použití při modelování firemních procesů a firemní struktury. Jako poslední bude prezentován *case* nástroj sloužící k vytváření grafických modelů podporující využívané syntaxe.

Praktická část

Na začátku této kapitoly bude představen postup, jak správně vytvářet modely. A jak s nimi zacházet, aby nedošlo k jejich znehodnocení. V další části budou zformovány tři modely. První bude znázorňovat ERP systém vývojové firmy. Druhý model bude vytvářet odraz firemních procesů výrobní firmy, tak aby ve třetím modelu šlo na dva předchozí navázat a mohl být vytvořen model odrážející firemní procesy výrobní firmy po naimplementování ERP systému. Průběh tvorby těchto modelů budou doprovázet diagramy podnikové struktury, jež budou mít za úkol získávat dodatečné informace, které budou důležité pro vytvoření a nasazení nového ERP systému do produkčního podniku.

3 Teoretická část

3.1 Výroba

Podle Kořenovského (2012) představuje výroba činnost, kterou společnost vykonává, tak aby mohla poskytovat výrobky nebo služby, z nichž získává od svých zákazníků peníze. Výstupem výroby může být tedy nejen hmatatelný výrobek, který si většinou lidé přestaví pod pojmem výroba, ale také služba, která má svůj specifický výrobní proces. Při pohledu na výrobu z širšího hlediska je možné zjistit, že se realizuje nejen v průmyslu, ale také v dopravě, poradenských firmách, školách, nemocnicích, úřadech atd. Kdo jejímu řízení a znalosti potřebných procesů nevěnuje dostatek pozornosti, může v určité chvíli zjistit, že konkurence je již daleko před ním. Výroba totiž velkou měrou rozhoduje nejen o nákladech, produktivitě, konkurenceschopnosti, spokojenosti zákazníků, ale také o zisku a podnikatelské úspěšnosti společnosti. Výrobu je tedy možné definovat jako transformaci produkčních faktorů do ekonomických statků a služeb, které jsou nadále spotřebovávány.

Pokud by se výroba zaměřovala pouze na ekonomická a společenská hlediska, dostala by se do situace, kdy jsou všechny výrobní zdroje využívány efektivně. Efektivnost výroby je jedním z nejdůležitějších faktorů pro úspěšnost podniku. Podmínkami tržní ekonomiky jsou firmy do jisté míry, zejména kvůli konkurenci, tlačeny k tomu, aby své výrobní faktory využívaly co nejefektivněji, neboli vyráběly co možná s nejnižší spotřebou výrobních faktorů. V dnešní době, kdy velkou část výroby zastávají stroje, je však velmi těžké získat náskok před konkurencí, aniž by nebylo nutné nakupovat nové a efektivnější stroje. Nicméně lze lépe analyzovat a poznat firemní procesy. A tak zlepšovat současný stav nejen z pohledu výroby, ale také díky vyšší znalosti výrobních procesů jednotlivými pracovníky (ManagementMania, 2016)

3.1.1 Charakteristika výrobních faktorů

Výrobní faktory, ke kterým patří práce, půda, kapitál a informace jsou nejzákladnějšími ekonomickými zdroji a v podobě statků a služeb představují

vstupy do ekonomických procesů. Tyto ekonomické zdroje jsou vzácné, neboť jejich výskyt je omezený, na rozdíl od lidských potřeb, které jsou neomezené. Neomezenost lidských potřeb vyvíjí tlak na výrobu, která spotřebovává ekonomické zdroje. Lidskými potřebami je tedy poháněn celý výrobní proces.

Nejčastěji se výrobní faktory diferencují na tři základní skupiny: půda (přírodní zdroje), práce, kapitál. Je však důležité, aby se nezapomínalo i na čtvrtou skupinu, kterou jsou informace. **Práce** zahrnuje veškeré lidské zdroje, které jsou uplatňovány ve výrobním procesu. Pro nastavení správných firemních procesů hraje nejdůležitější roli kvalita managementu. Pojem **půda** je označován jako souhrn všech součástí přírodního prostředí: např. orná půda, lesy, zdroje nerostných surovin, voda, vzduch. **Kapitál** je jediným produkčním faktorem, který je sekundární. Vzniká tedy v průběhu výroby a je uplatňován jako vstup do další produkce. Tímto znakem se kapitál liší od půdy a práce, u nichž se předpokládá, že nemohou být výsledkem výroby. (Kucharčíková, 2011) Jako čtvrtý faktor jsou zmiňovány **informace**, které jsou nezbytné pro inovace výrobních procesů. Bude-li však společnost pouze sbírat velké objemy údajů a nebude-li s nimi umět patřičně zacházet, dojde velmi rychle k jejímu zahlcení velkým množstvím dat. Ta pak budou pro podnik drahá a neefektivní. (ManagementMania, 2017)

Zde je patrné, že pro firmu jsou nezbytné všechny zmíněné výrobní faktory. A není možné maximalizovat zisk a zamezit plýtvání některých z těchto faktorů, aniž by firma neznala výrobní procesy a metody, které využívá.

3.1.2 Přístupy v řízení výroby

V průběhu let byly v průmyslově vyspělých zemích postupně vyvíjeny metody pro správné řízení výroby, které vedou ke zvýšení její efektivnosti. Tyto metody vycházejí z určitých principů a filozofických přístupů k výrobnímu managementu, které byly realizovány a uznávány ve své době. Jejich společným znakem je, že vznikaly především za účelem eliminace neefektivnosti a plýtvání dříve používaných metod v řízení výroby. Proto, aby zainteresované strany měly představu, jak výroba funguje, je nezbytné, aby znaly nejen procesy v běžném

chodu výroby, ale také aby ovládaly metody jejího řízení, které napomáhají právě k optimalizaci fungování výroby.

Material Requirement Planning (MRP)

MRP (plánování požadavků materiálu) je způsob řízení, který je zaměřen spíše na management zásob materiálu, nežli na plánování a řízení průběhu výroby. Hlavním účelem MRP je nahradit do té doby využívané řízení zásob. MRP je normovaný efektivnější způsob, který je zaměřen na adresné objednávání materiálu podle skutečných potřeb výroby, kde požadované informace jsou získávány prostřednictvím výpočetní techniky. Pro výpočet plánovaných potřeb materiálu neboli analýzu MRP se používá hrubý rozvrh výroby. Jedná se o výrobní plán pro jednotlivé časové intervaly, nejčastěji týden, pro které se stanoví počet výrobků, které musí být dokončeny. Tento plán je sestaven na základě objednávek nebo předpovědi poptávky po výrobcích. Důležité je, aby byl do výpočtů zahrnut disponibilní materiál nacházející se ve skladu. Výhodou tohoto přístupu je snížení nákladů na pořizování a údržbu zásob. Nevýhodou je, že plánování spočívá pouze na informacích vycházejících z hrubého rozvrhu výroby a nikoli ze skutečného průběhu výroby, což může vést v případě odchýlení od plánu ke zbytečnému zvyšování zásob či jejich nedostatku. (Kavan, 2002; Keřkovský, 2012)

Manufacturing Resource Planning (MRP II)

MRP II (plánování výrobních zdrojů) vychází ze systému řízení MRP, má však výraznější propojení objednávek materiálu s podrobným rozvrhnutím výroby a kapacitními výpočty. Tento přístup byl vyvinut v 70. letech a v mnoha podnicích je používán dodnes. Hlavním přínosem MRP II je snížení závislosti na oběžných prostředcích, která je hlavním problémem řízení výroby v podnicích. Snížení závislosti tak vede ke zvýšení úspory nákladů vynaložených na pořizování a údržbu zásob. V dnešní době bývá MRP II součástí většiny programových systémů pro řízení výroby. MRP II je v podstatě přístup řízení MRP, které je doplněno o podrobnější plánování výroby a kapacitních propočtů umožňující i vazby na řízení prodeje. Nedostatkem MRP II jsou nedostatečná vstupní data, zejména odhady pracnosti jednotlivých úkolů. Dále jsou také komplikovanější výpočetní techniky na rozdíl od MRP. (Keřkovský, 2012)

Enterprise Resource Planning (ERP)

V systémech typu MRP II je důležité zabezpečit nejen informační řízení výrobních procesů, ale i veškeré informace potřebné pro rozhodování. Proto jsou z těchto systémů vytvářeny cestou integrace nové subsystemy, kterými jsou informační systémy ERP (Enterprise Resource Planning). Základem ERP je databáze, která je společná pro všechny napojené oblasti jako je obchod a marketing, distribuce, technologie, finance, účetnictví, CRM, řízení lidských zdrojů atd. Z pohledu informačního systému lze ERP chápat jako komplexní softwarový balíček, který umožňuje společnosti účelně a efektivně řídit podnikové zdroje (např. finanční zdroje, investiční majetek, zásoby apod.). V porovnání s dřívějšími trendy, kdy každá oblast firmy byla řešena samostatnými aplikacemi, ERP umožňuje jedním systémem pokrýt veškeré potřeby organizace, protože integruje veškerá data a procesy organizace do jednotného celku. Nejčastěji ERP systém vyžívá k dosažení integrace řadu softwarových modulů a hardwarové infrastruktury. Hlavní přínosy ERP systémů jsou v automatizaci a integraci základních firemních procesů, sdílení společných dat a jejich zpracovávání v rámci celého podniku, a to v reálném čase. ERP systém zjednodušuje plánování, řízení výroby, řízení zakázek atd. Automatizace zakázek umožňuje snížení časové prodlevy od pořízení dat po jejich zpracování. Velké přínosy toto řešení přináší ve firmách, které mají jednotlivé části nebo pracovníky od sebe vzdálené. Díky společnému softwaru mohou společně sdílet vždy aktuální informace. (ManagementMania, 2016, Keřkovský, 2012)

Optimized Production Technology (OPT)

Koncept řízení výroby OPT byl vyvinut v 70. letech. Na rozdíl od MRP II se zaměřuje na optimalizaci výrobních toků (např. průchod součástí, výrobků nebo zákazníků napříč výrobním systémem), tak aby byla co nejlépe využita pracoviště s procesem, který má omezenou výrobní kapacitu a omezuje tak ostatní procesy výroby (tzv. bottlenecks). OPT je založen na myšlence, že výkonnost celého výrobního systému je založena na nejslabším místě (procesu) firmy. Na OPT je možné pohlížet z více hledisek, tedy jako na nástroj pro řízení výroby, nástroj na zlepšování organizace výroby, nebo jako dokonalý software pro plánování výroby. Za hlavní přínos OPT se bere schopnost redukovat průběžné doby a celkové

zvýšení průchodnosti výrobním systémem. Z toho je patrné, že OPT je více přizpůsoben dynamickým podmínkám nežli nákladově orientované koncepty jako MRP nebo MRP II. Další výhodou OPT je možnost zadávání parametrů do systému, díky čemuž má uživatel široké možnosti přizpůsobovat výrobní procesy konkrétním podmínkám a zohledňovat tak konkrétní specifika (např. organizaci výrobních procesů, jejich parametry, cíle a kritéria výrobního plánování). Jako kritické faktory OPT lze označit potřebu co nejpřesnějších vstupních dat a náchylnost k jejich změnám. Dle přesnosti dat se pak odvíjí doba trvání jednotlivých operací a také údaje o disponibilních výrobních kapacitách.

Just-in-time (JIT)

Metoda JIT byla vytvořena počátkem 80. let a poté i uplatněna v řízení výroby v Japonsku, USA a západní Evropě. Základní myšlenkou JIT je výroba pouze nezbytných položek v požadované kvalitě, v nezbytném množství a v co možná nejpozději přípustném čase. Hlavním úkolem JIT je eliminovat pět základních druhů ztrát, kterými jsou nadprodukce, čekání, doprava, udržování zásob a nekvalitní výroba. Implementaci JIT je nutné brát jako významný strategický záměr, který musí vycházet především z výrobní strategie firmy a být s ní v souladu. Nesmí se však zapomínat na ostatní strategie, a především na to, že firma je celek a ovlivnění jedné její částí ovlivní celou společnost. JIT používají podniky, pokud chtějí minimalizovat dopravní a skladovací náklady, ovšem zavedení JIT klade vysoké nároky na naprosto přesné řízení všech procesů a toků, kterých se daná metoda týká. Za hlavní přínosy této metody je bráno snížení zásob a zvýšení efektivnosti výroby tím, že se eliminují rozpracované výrobky. Dále by mělo dojít k zredukování výrobních a skladových prostor. Důsledkem by měly být, kratší seřizovací časy, vyšší využití výrobních zdrojů, vyšší produktivita a snížení režijních nákladů. Naopak nevýhodou této metody je díky důrazu na minimální zásoby možnost, že je společnost příliš závislá na dodavatelích a v případě jejich výpadku nemůže vyrábět. JIT také vyžaduje velké nároky na dopravu a informace potřebné k plánování. Samotné zavádění metody vyžaduje poměrně velké náklady a jejich návratnost může být až v řádu několika let. (Keřkovský, 2012; ManagementMania, 2016)

Kanban

Jedná se o flexibilní a samoregulační systém řízení výroby na principu JIT. Hlavním prvkem této metody jsou nosiče informací kanbany (japonsky označované štítky), které plní roli objednávek a průvodek. Pro objednání určitého dílu, je kanbanů jen omezené množství a toto množství odpovídá povolené úrovni zásob rozpracovaných dílů a výrobků. Pracoviště, na kterém dochází zásoba potřebného materiálu určitého druhu, odešle objednávkový kanban společně s přepravním vozíkem na pracoviště, které potřebné součástky dodává. Toto pracoviště vozík naplní požadovaným množstvím a společně s průvodním kanbanem jej odešle zpět na pracoviště, které si je objednalo. Objednávané množství bývá většinou velmi malé např. jedna desetina z celkové denní potřeby. O dodávku tedy vždy žádá pracoviště, které je ve výrobním procesu dál a pracoviště, které objednávku vyřizuje jí musí splnit vždy v požadovaném množství a čase. V případě, že na pracoviště dorazí více objednávek, jsou tyto objednávky vyřizovány metodou FIFO, tedy první je vyřízena ta, která přišla dříve. Každá ze zásilek nejenže musí obsahovat požadované množství, ale všechny odeslané součástky musí odpovídat požadované kvalitě. Pokud některá ze součástí neodpovídá, musí být opravena nebo vyřazena dříve, než je objednávka odeslána. Potřeba změny množství zásob je řešena pomocí navýšení kanbanů v oběhu. Kanban je vhodný pro opakovanou výrobu. (Tomek, 2007; Horváth, 2007)

Štíhlá výroba (lean management)

Štíhlá výroba je v dnešní době vnímána jako koncept, filozofie, praxe a soubor nástrojů v jednom. Na rozdíl od silně centralizovaného řízení, které se zaměřuje na nízké náklady a vysokou produktivitu a kde individuální požadavky zákazníků nepatří mezi nejdůležitější priority, se štíhlá výroba snaží pružně reagovat na požadavky klienta a poptávku. Ta je řízena decentralizovaně prostřednictvím pracovních týmů, při minimalizaci zbytečných výrobních kroků. Každý ze zaměstnanců nese odpovědnost za požadovanou kvalitu a průběh výroby. V systému štíhlé výroby jsou kompetence pracovníků decentralizovány a v případě zjištění chyby ve výrobním procesu může každý z pracovníků výrobu přerušit. Štíhlá výroba se snaží zaměřit na maximální kvalitu, minimalizaci zbytečných

kroků a zvyšování hodnoty produktu tím, že se bude orientovat na maximální uspokojení potřeb zákazníka, což je přímým opakem tradiční hromadné výroby. Jeden z principů štihlé výroby je, že pracovníci nevyrábí v tradičním systému jeden výrobek za druhým, ale čekají na požadavek od následujícího pracovního procesu, dokud neobdrží požadavek na určitý počet potřebných výrobků. Jednotlivá pracovní oddělení se tak mezi sebou stávají interními zákazníky, kdy první výrobní stupeň musí splnit všechny požadavky a uspokojit tak potřeby následujícího výrobního stupně. Hlavní předností systému ve výrobě je, že nevznikají náklady v důsledku vytváření mezioperačních zásob a je zkrácena průběžná doba výroby. (Keřkovský, 2012; Dombroski, 2013)

Štihlá výroba je metoda zaměřená na optimalizaci procesů, ale také na co možná největší uspokojení potřeb zákazníka. Jedná se tedy o to, aby docházelo ke správnému plánování a kontrole spotřeby všech výrobních faktorů a tím se zabránilo plýtvání. Aktivita v systému štihlé výroby se posuzují především podle toho, zda jsou schopny vytvářet hodnotu, kterou je ochoten zákazník zaplatit. Aktivita, které nejsou schopny vytvářet přidanou hodnotu, ale přesto se ve firmě nacházejí, ukazují na zbytečné plýtvání. Příkladem takovýchto aktivit mohou být např. opravy nekvalitně provedené práce, skladování dílů mezi navazujícími výrobními procesy, několikanásobná evidence dat, dlouhé dopravní cesty uvnitř podniku a s tím spojená ztráta času při zbytečném čekání na materiál a nadbytečné udržování zásob. Takovéto aktivity je možno odhalit nejen u činností bezprostředně spojených s výrobou, nákupem, vývojem, ale také ve správě managementu. I v případě, že jsou tyto aktivity nalezeny a redukovány, tak zlepšování firemních procesů není u konce. Firma, která zkvalitní své procesy a dále pak poleví ve zlepšování, dělá chybu, když se domnívá, že dosažený bod je dostačující. Metoda štihlé výroby je nekonečným procesem, který probíhá kontinuálně a stále hledá místa, ve kterých by mohlo dojít ke zlepšení. Toto neustálé zlepšování je důležité pro udržení konkurenceschopnosti. Firmy často dělají chybu, že v obdobích velkého počtu zakázek mají tendenci se spokojit s dosaženými výsledky, tento stav však v budoucnu vede k neúspěchu. (Keřkovský, 2012; Jurová, 2011)

Jako další princip štíhlé výroby je zaměření na podstatné aktivity společnosti, které přinášejí největší zisk a ve kterých je společnost dobrá. Omezení společnosti na podstatné aktivity však vyžaduje posouzení rozsahu působnosti všech útvarů firmy a analýzu prvků, které podnik ovládá lépe než konkurence nebo externí partneři a také co nejvíce přispívá ke zlepšení konkurenční výhody z hlediska zákazníků. Štíhlá výroba pak přikazuje zaměřit všechny interní kapacity a zdroje podniku na využívání a zlepšování právě těchto klíčových vlastností firmy. (Keřkovský, 2012; Dombroski, 2013)

3.1.3 Industry 4.0

Samostatnou a v dnešní době hodně řešenou kapitolou je Industry (průmysl) 4.0. Jedná se o pokročilou strategii německé vlády zaměřenou na automatizaci průmyslu. Ten je založený na kyberfyzikálních systémech (tento systém se skládá z fyzických entit, které jsou řízeny počítačovými algoritmy, základem je spolupráce samostatných výpočetních jednotek, které jsou schopny se autonomně rozhodovat) nasazovaných do zařízení používaných v každodenních oblastech života. Tímto se především odlišuje průmysl 4.0 od obyčejné automatizace produkčních systémů, a proto je také nazván již 4 průmyslovou revolucí/evolucí. Základem je internet věcí (IoT), který má umožňovat propojení nejrůznějších zařízení pomocí internetu, což umožní otevřít nové možnosti ovládání, monitorování, komunikování a propojení domácích zařízení, aut, ale také medicínských zařízení. Pro nasazení tohoto systému je nutné, aby všechna výrobní zařízení obsahovala integrované komunikační standardy, přes které bude CPS s těmito zařízeními komunikovat. Průmysl 4.0 vymezuje pojem digitální továrna na inteligentní továrna, která je schopná adaptace, efektivně využívá zdrojů, ergonomické uspořádání (vyhovující nárokům člověka) a integrování zákazníků a obchodních partnerů do procesů podniku. Příchod IoT umožňuje přechod od sériové výroby na výrobu orientovanou na zákazníky. Výroba probíhá v malých dávkách a individuální produkci, přičemž nedochází k nárůstu ceny výrobků. Automatické stroje a ostatní chytré nástroje využívané v industry 4.0 společně bezdrátově komunikují s IT systémy, která mají cloudové řešení. Spojení fyzických zařízení s jejich virtuálními daty vede nejen ke zlepšení výrobních procesů, ale také

k změně hodnotového řetězce od návrhu výrobky přes jeho výrobu a logistiku, až po jeho recyklaci. Pro zavedení průmyslu 4.0 bude nutné provést změnu většiny stávajících podnikových procesů od vývoje produktů až po jejich pozáruční servis. Avšak nové inteligentní továrny za pomoci IoT budou moci nejen zlepšit kvalitu a návaznost jednotlivých procesů a tím i produktů vyrobených přesně na základě individuálních požadavků zákazníků, ale také bude lepší komunikace se zákazníkem. Ten nejenže bude moci sledovat stav svého výrobku, ale také společnost může sledovat prediktivní údržbu a tím i optimalizovat své výrobní kapacity. (Jurová, 2016)

3.2 Procesní řízení

Hlavním stavebním kamenem procesního řízení je proces, a proto je důležité nejdříve objasnit co to takový proces je. Každý člověk má určité aktivity nebo rituály, které provádí každý den a jsou si velice podobné. Příkladem mohou být ranní činnosti. Každý z jedinců poté co vstane, provádí rozdílné činnosti. Někdo se nejdříve musí umýt jiný, zase nasnídat. Ovšem zaměříme-li se na konkrétního jedince, zjistíme, že svůj den začíná vždy velmi podobným způsobem (např. vždy po ránu si jde uvařit kávu). Tato činnost, kterou je vaření kávy se v managementu a také procesním řízení nazývá „proces“. Každý jedinec, aniž by si to uvědomoval, provádí během dne soustavu aktivit, kterými jsou právě zmíněné procesy. Člověk nejenom, že může dané procesy vytvářet, ale může se také podílet na procesech někoho jiného. Své procesy může poskytovat ostatním nebo může využívat výhod poskytující procesy ostatních, kterými jsou služby nebo produkty (např. káva zakoupená v kavárně). Z toho vyplývá, že každý prožívá svůj život v systému jednotlivých procesů, které na sebe navzájem navazují, které se doplňují nebo si mohou odporovat. A nezáleží na tom, jestli se jedná o procesy systémové nebo nesystémové. Aby však procesy mohly být systémové, navzájem na sebe navazovaly, aby se doplňovaly a vytvářely ze vzájemného působení nějaký užitek, musí být řízeny. (Fišer, 2014)

Procesní přístup vychází z podmínky, že základním objektem řízení je popsáný, jasně definovaný, strukturovaný, zdrojově a vstupy zajištěný proces, který je vytvářen pro konkrétního zákazníka a mající jednoznačně stanoveného vlastníka.

Úkolem každého z procesů je poskytnout výrobek nebo službu určitému zákazníkovi s ohledem na jeho definované požadavky, stanovená pravidla a omezení. Proces vyžaduje určité vstupy a může využívat zdroje, které mu jsou přiřazeny. Proces je tedy nějaký soubor činností, které na sebe vzájemně působí a které přeměňují vstupy na výstupy. Pro lepší pochopení procesu bude vhodné uvést příklad již zmíněné přípravy kávy. Vezměme si tedy osobu, která si chce připravit kávu. Pro přípravu kávy jsou potřeba nějaké suroviny. V tomto případě to budou vstupy, kterými jsou káva a voda. Jako další vstupy mohou sloužit mléko, nebo cukr odvíjející se podle požadavků osoby. Aby mohly být tyto vstupy proměněny ve výstup (kávu), je nezbytné využít potřebné zdroje. V tomto případě mohou být zdrojem rychlovarná konvice, elektrická energie, hrníček a osoba vykonávající potřebné aktivity k uvaření kávy. Z toho vyplývá, že vstupy jsou něco, co tvoří přidanou hodnotu a zdroje slouží k tomu, aby ze vstupů mohla vzniknout přidaná hodnota. Podívá-li se čtenář na tento proces a rozebere-li si ho po jednotlivých aktivitách, získá jednotlivé činnosti, které na sebe navazují: 1. Uvaření vody v rychlovarné konvici. 2. Nachystání kávy do hrníčku. 3. Zalití kávy horkou vodou. 4. Doplnění kávy o cukr a mléko dle požadavků zákazníka. 5. Naservírování kávy.

Nyní je vidět, pokud se vezme i takto jednoduchý proces a popíše se podle jednotlivých charakteristik, že vyvstane najednou na povrch mnoho dalších okolností, které dosud nebyly zjevné, nebo si je jen dotyčný člověk v rámci rutinní činnosti ani neuvědomoval. Vezmou-li se daleko složitější a rozsáhlejší procesy, které se podobným způsobem rozeberou, objeví se celá řada chyb, nedostatků, neefektivnosti nebo nehospodárnosti, kterou bylo dříve velice těžké nebo nemožné identifikovat, jednoznačně popsat a odstranit. Toto vše však nyní umožní procesní přístup. (Grasseová, 2008)

Za zmínku jistě stojí i funkční přístup k řízení společnosti, který je již zastaralý a neefektivní, ale mnoho firem ho dnes neustále hojně využívá a kvůli nastoleným zvyklostem ani nechce opustit. Funkční přístup spočívá v dělbě práce, při které je práce rozložena na nejjednodušší úkony, tak aby je mohli provádět i nekvalifikovaní pracovníci vedení určitým počtem specialistů. Tento přístup vedl k zavedení hromadné výroby a rozdělení práce mezi funkční jednotky vytvořené

na základě jejich dovedností (odbornosti). Tomuto členění také odpovídají organizační struktury, kdy je firma rozdělena na divize a jednotlivé útvary, kde každý z útvarů vykonává pouze své úkoly, ale už nesledují celý tok podnikových procesů. V případě zmíněného funkčního přístupu představuje každý přechod mezi procesy, mezi útvary, rizikové místo z hlediska informační bariéry a časové ztráty. Organizace je pak řízena potřebami jednotlivých útvarů, a aby mohlo dojít k navýšení produktivity jako celku, musí být zvýšena produktivita u každé funkční jednotky zvlášť. Dnes však už do velké míry zasahují do výroby zákazníci a produkce se upravuje podle jejich požadavků. Funkční přístup dnes již není efektivní a do budoucna pro firmy nebude ani udržitelný. (Grasseová, 2008; Carda, 2001)

Oproti funkčnímu přístupu, který klade hlavní důraz na organizační dělení dovedností, je procesní přístup k řízení orientovaný nejen na výsledek práce (produkt), ale také na sled aktivit, které vedou k dosažení daného produktu. Práce není vykonávána v jednotlivých odděleních, které jsou od sebe navzájem odděleny, ale práce skrz tato oddělení protéká. Celý systém je pak řízen potřebami zákazníků. Při využití procesního přístupu dochází k postupnému vylepšování jednotlivých procesů a tím i k jejich optimalizaci a zjednodušení celého toku práce.

Jak z názvu, tak z textu vyplývá, že procesní řízení je založeno na firemních procesech. K přechodu na tento styl řízení je pro společnost nezbytné, aby znala své primární procesy a byla odborníkem v oblasti, které se věnuje, protože společnost, která nezná své procesy, nemůže chtít ani tyto procesy zlepšovat. Když jsou naopak známy hlavní procesy, je možné odstranit nadbytečné procesy a zaměřit síly společnosti na ty hlavní, které budou dále rozvíjeny. (Fišer, 2014)

3.3 BPMN

Business Process Modeling and Notation (BPMN) je grafická notace sloužící k popisu firemních procesů. Primárním cílem BPMN je poskytnout notaci, která je snadno srozumitelná business útvarům i IT oddělení. Základem BPMN je vytvoření popisu posloupnosti činností ve společnosti, včetně událostí provázejících proces nebo komunikace mezi jednotlivými subjekty. V současné době se jedná o standard pro grafické zobrazení analýzy procesů.

3.3.1 Vývoj BPMN

BPMN byla původně vytvořena neziskovou společností Business Process Management Initiative (BPMI), která se nejprve snažila vytvořit standard obchodních procesů jako prostředek rozvoje e-byznysu a B2B. Cílem BPMI je snaha podporovat a rozvíjet využívání Business Process Managementu (BPM je systematický přístup, který se snaží dosáhnout účinnějšího procesu práce v organizaci a také možnost účinněji a schopněji se přizpůsobit stále se měnícímu prostředí) prostřednictvím vytváření standardů pro návrh, zavádění, uskutečňování, údržbu a optimalizaci procesů. Tímto způsobem zamýšlela společnost BPMI podnikům usnadnit vzájemné ovlivňování na trhu a podporovat další rozvoj na globálním trhu. Neboť na trhu nebo dokonce ve společnosti se vyskytovaly stejné procesy, které však byly jinak charakterizovány a docházelo tak k iluzi, že se jedná o rozdílné procesy, čemuž chtěla společnost BPMI zabránit a umožnit organizacím efektivněji komunikovat a sdílet nejen data, ale i aplikace. (Douglas, 2005)

V předchozím odstavci se objevilo několik zkratk, které mohou být pro někoho neznámé a jelikož v rámci BPMN je těchto zkratk víc bylo by vhodné si je nejprve objasnit. Aby mohly být vytvářeny grafické modely, vznikl jazyk BPML (Business Process Modeling Language) založený na syntaxi XML, který měl definovat stejná pravidla pro všechny zúčastněné strany. Stal se tak meta-jazykem pro modelování obchodních dat. Tento jazyk byl otevřený a každý si ho mohl stáhnout ze stránek BPMI. Jelikož však bylo nutné vytvořit k tomuto jazyku notaci, která bude pro všechny využívající tento jazyk snadná a srozumitelná, vznikla notace BPMN. BPMN je tedy jakýmsi standardem pro jazyky na bázi XML (např. BPML). (BPMI.org, 2002; Douglas, 2005; A. White, 2012)

Vznik BPMN

Notace BPMN vznikla již v roce 2002 ve verzi 0.9 jako koncept, který by měl poskytovat právě zmíněnou grafickou notaci, která bude jednoduchá, ale přesto komplexní a srozumitelné pro všechny, kteří s ní budou pracovat. (BPMI, 2002)

Vydání veřejné verze BPMN

O dva roky později v květnu 2004 byla vydána první verze syntaxe 1.0, která byla dostupná pro veřejnost. (A. White, 2012)

V roce 2006 došlo ke spojení společnosti BPMI a společnosti OMG (Object Management Group), která nyní syntaxi BPMN spravuje.

V roce 2008 byla vydána další verze s označením 1.1, která měla přinést některé drobné změny. Hned o rok později vyšla další aktualizace syntaxe s označením 1.2, která obsahovala některé opravy a upřesnění, které vznikly v předchozí verzi. (OMG, 2008; OMG, 2009)

BPMN 2

V lednu 2011 byla vytvořena verze 2.0, která přinášela mnoho rozšíření a také byla považována za přelomovou. Původní elementy však neprošly změnami a díky tomu zůstala zpětná kompatibilita s předchozí verzí 1.2. Nově tak přibyla další rozšíření, jako je například Ad-hoc aktivita, která umožní, že vnořené aktivity se mohou spouštět v libovolném pořadí. Multiple Instances označuje vícenásobnou aktivitu, která může být vykonávána paralelně nebo sekvenčně a mnoho dalších, které budou popsány v následující kapitole.

O rok později v roce 2012 byla představena dosud poslední verze BPMN 2.0.2. (OMG, 2014)

Ač BPMN do dnešní doby ušlo velkou vzdáleností a má již určité předpisy, jak má být jazyk používán. To také udává ISO/IEC 19510:2013. Syntaxe je však stále docela otevřená a dovolí architektovi, aby si jí částečně upravoval podle svých potřeb nebo svého chápání. V takovém případě mohou vznikat drobné nuance, které však mohou tvořit bariéry mezi uživateli komunikujícími mezi sebou pomocí tohoto jazyka. Proto je vhodné tvořit celopodnikové metodiky, které budou napomáhat v zaškolování nových pracovníků, ale především k jednotným modelům v rámci společnosti. (ISO, 2013)

3.3.2 Elementy syntaxe BPMN

3.3.2.1 Flow Objects (Tokové objekty)

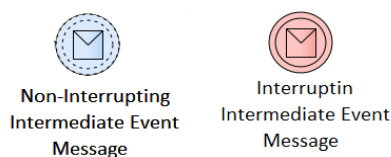
Event (Události)

Událost je jedním ze třech základních elementů tokových objektů. Tyto události ovlivňují tok modelu a obvykle spouští (Start), specifikují (Intermediate) nebo ukončují (End) celý proces. Mimo toto rozdělení do těchto tří kategorií se můžou události dále dělit na ty, které čekají na přijetí a ty, které něco odesílají. Spouštějící události (Start Event) mohou reagovat pouze na „Catch“, tedy „chytit“ a znamená to, že čekají na přijetí nějaké události. Pro znázornění této události se používá obrys znaku dané události. Naproti tomu ukončující události (End Event) mohou reagovat pouze na „Throw“, tedy „hod“, což znamená, že tato událost něco odesílá. Událost odeslání je znázorněna pomocí černě vyplněného znaku dané události. Specifikující událost (Intermediate) pak může být v pozici odesílající události i události, která čeká na přijetí. (OMG, 2014)



Obrázek 1 Druhy událostí (zdroj: [17], vlastní úprava)

Posledním rozdělením, kterého mohou události nabývat je schopnost přerušit či nepřerušit danou aktivitu. Přerušování „Interrupting“ modeluje událost, která je schopna přerušit daný sub-process, který byl dříve vyvolán jinou událostí a s ním i její tok. Pokud je však potřeba vytvořit událost, která nepřerušuje dříve spuštěný tok, použije se „Non-Interrupting“ událost. Tyto události se mohou modelovat do procesů jako Start nebo Intermediate a jejich spuštění nezpůsobí ukončení procesu, ale vytvoří nový tok, který běží souběžně s již spuštěným. Jak z tohoto textu vyplývá, Interrupting a Non-Interrupting události se používají vždy ve spojení s aktivitou (sub-procesem), který buď ukončí nebo v něm vyvolají nový tok. Pro popis událostí se používá sloveso v minulém čase. (OMG, 2014)

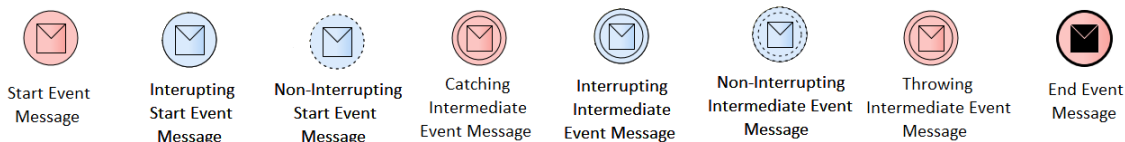


Obrázek 2 Non-Interrupting a Interrupting Event (zdroj: [17], vlastní úprava)

Message (Zpráva)

Event Message představuje spouštěč, který ve chvíli, kdy obdrží zprávu, zahájí proces. Proces nemusí nutně reprezentovat nějaký počítačový systém, ale může se jednat např. o výrobní systém. Nezáleží tedy, jestli se jedná o elektronickou zprávu nebo tištěnou. Zpráva nejčastěji představuje komunikaci mezi dvěma účastníky. Message událost může nabývat všech výše zmíněných rozdělení. Událost může být propojována pomocí vazby Message flow, která určuje i směr toku. (OMG, 2014)

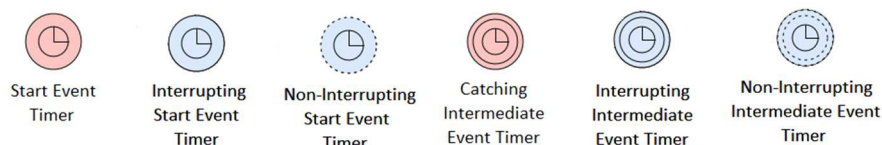
Na obrázku níže je možné vidět výčet všech událostí typu message. Zajímavé jsou hned dvě první události, které se podle notace tváří úplně stejně, jejich rozdíl je však v použití. Jak již bylo zmíněno výše, klasický Start Event se používá vždy na začátku procesu, aby jej spustila, kdežto Interrupting Start Event je vždy spojen s konkrétní aktivitou (resp. sub-procesem). Dále tu jsou zprávy typu Catching, které jsou vyvolány nejčastěji jiným procesem a zprávy typu Throwing, které naopak spouští některý další proces. Tyto další procesy nemusí být modelovány. V případě, že však jsou modelovány, dochází k propojení jednotlivých procesních diagramů.



Obrázek 3 Zpráva (zdroj: [17], vlastní úprava)

Timer (Časovač)

Časovač je událost, která představuje konkrétní časový okamžik nebo časový cyklus. Událost pak v určitý časový okamžik spustí proces. Takový časovač může být např. událost, která každý den spustí stroje před zahájením výroby, nebo pravidelná kontrola materiálu na skladu. Časovač může být využit jako součást nějaké aktivity. V případě, že do určité doby není aktivita splněna, tok pokračuje alternativní cestou (např. aktivita čekající na vyjádření zákazníka, pokud se zákazník nevyjádří do stanoveného času, je mu zasláno upozornění). (OMG, 2014)



Obrázek 4 Časovač (zdroj: [17], vlastní úprava)

Error (Chyba)

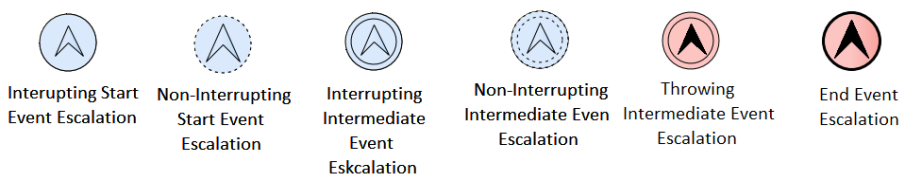
Událost Error vzniká, vyskytne-li se nějaká známá chyba v chování. Na tuto událost by měla vždy být vytvořena aktivita, která nabídne uživateli pomoc nebo ho alespoň s danou chybou seznámí. Tato událost může být využívána pouze ve spojení s nějakou aktivitou a může nabývat pouze stavu interrupting nebo end. Příkladem této události může být požadavek zaměstnance na zaslání objednávky zákazníkovi, ten však dosud nebyl zadán do systému, a tak tento systém vyhodí chybovou hlášku o tomto stavu a nabídne možnost zavedení zákazníka do systému. (OMG, 2014)



Obrázek 5 Chyba (zdroj: [17], vlastní úprava)

Escalation (Eskalace)

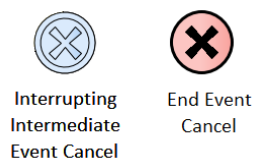
Eskalace je událost, která se může používat pouze u sub-procesů. Je to z toho důvodu, že v případě eskalační události je současná aktivita eskalována (ukončena a poslána na aktivitu vyšší úrovně), kde pokračuje tok, jako by byl celý sub-proces dokončen s tím rozdílem, že některé aktivity sub-procesu mohly zaniknout. Příkladem může být objednávka zboží, kdy se nejprve zjistí, zda je zboží na skladě a pokud není, je nutné provést objednávku u dodavatele. Pokud však zboží je na skladě, je možné tento krok přeskočit a aktivita může eskalovat přímo na odeslání objednávky. (OMG, 2014)



Obrázek 6 Eskalace (zdroj: [17], vlastní úprava)

Cancel (Ukončení)

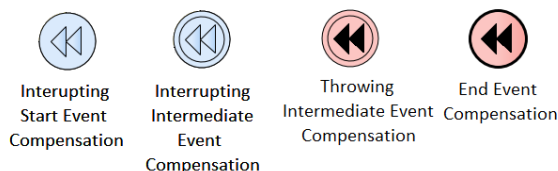
Tato událost se používá u transakčních sub-procesů, pro znázornění chování, které bude spuštěno v návaznosti na ukončený sub-proces. Příkladem může být rezervace hotelu, která se bude platit pomocí platební karty. V případě, že platba proběhne úspěšně, je hotel úspěšně zarezervován. V případě, že je karta odmítnuta, je spuštěna chybová hláška a rezervace je neúspěšně ukončena. (OMG, 2014)



Obrázek 7 Ukončení (zdroj: [17], vlastní úprava)

Compensation (Navrácení)

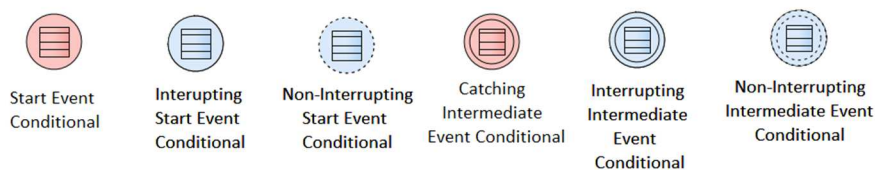
Událost Compensation se využívá transakčních aktivit k modelování požadavku navrácení. To znamená, že tato událost umožňuje návrat toku do určitého stavu. K tomu dojde i v případě, že byla aktivita zrušena, pak je tok vrácen do míst, kde došlo ke zrušení. Tato událost je podobná situaci, kdy je na webových stránkách dáno šipka zpět, která vrátí obsah o jeden krok zpět. (OMG, 2014)



Obrázek 8 Navrácení (zdroj: [17], vlastní úprava)

Conditional (Podmínka)

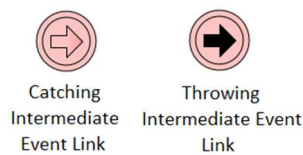
Tato událost je spuštěna ve chvíli, kdy je spuštěna určitá podmínka, jako v případě, že ve výrobní hale začne hořet a detektor kouře aktivuje požární poplach. Dalším příkladem může být např. snížením stavu zásob na skladu pod 10 %, tím dochází k odeslání požadavku na objednávku. (OMG, 2014)



Obrázek 9 Podmínka (zdroj: [17], vlastní úprava)

Link (Odkaz)

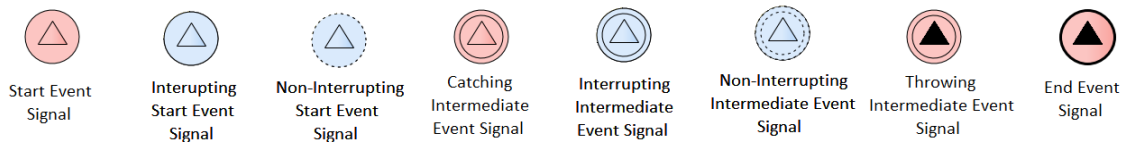
Událost link slouží k propojení více aktivit mezi sebou, aniž by musely být přímo propojeny vazbou. Jedná se o nahrazení sekvenčního toku v případě, že by došlo ke křížení vazeb a tím i ke snížení přehlednosti modelu. (OMG, 2014)



Obrázek 10 Odkaz (zdroj: [17], vlastní úprava)

Signal (Signál)

Tato událost vytváří signál, který je vyslán některým z procesů za účelem spuštění jiného procesu, který se může nacházet na jiném místě systému. Signál se liší od události Zpráva především tím, že nemá určitý cíl. Signál vyslaný jedním procesem může spustit jeden nebo více dalších procesů. (OMG, 2014)



Obrázek 11 Signál (zdroj: [17], vlastní úprava)

Terminate (Ukončení)

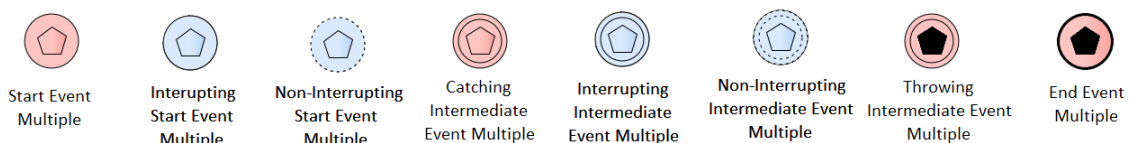
Tato událost reprezentuje bezprostřední ukončení veškerých aktivit uvnitř procesu, ale také ukončení celého procesu. Příkladem může být výrobní havárie, která ukončí celý proces výroby. (OMG, 2014)



Obrázek 12 Ukončení (zdroj: [17], vlastní úprava)

Multiple (Vícenásobný spouštěč)

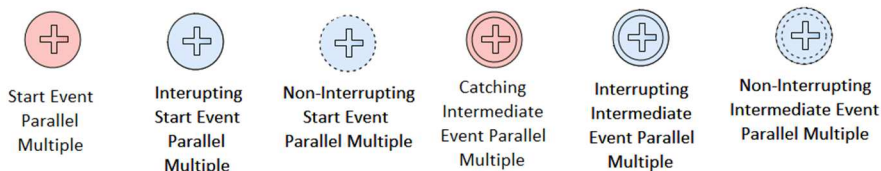
Událost Multiple je definována jako spouštěč, který nabízí několik způsobů, jak daný proces spustit. K spuštění pak stačí, aby byl splněn pouze jeden ze způsobů. Jednotlivé způsoby musí být u události popsány. Příkladem může být událost objednání jídla, které může být způsobeno reklamou v televizi nebo doporučením přítele. (OMG, 2014)



Obrázek 13 Vícenásobný spouštěč (zdroj: [17], vlastní úprava)

Parallel Multiple (Paralelní Spouštěč)

Tato událost je podobná události Multiple, také obsahuje několik způsobů (podmínek), které jsou nabízeny pro spuštění procesu s tím rozdílem, že musí být splněny všechny požadavky na tuto událost a až poté může tok pokračovat k následujícímu procesu. (OMG, 2014)



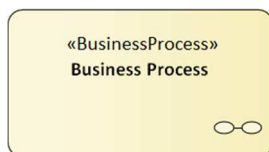
Obrázek 14 Paralelní Spouštěč (zdroj: [17], vlastní úprava)

Activities (Aktivity)

Aktivity představují určitou činnost, která je vytvářena v procesu firmy. Jednotlivé aktivity na sebe navazují pomocí vazeb sequence flow a vytvářejí ucelený proces. Aktivity se dále dělí na dva typy, kterými jsou sub-procesy a tasky. Sub-procesy jsou složitější aktivity, které mohou obsahovat více dílčích aktivit. Druhým typem jsou tasky, které mohou být součástí sub-procesu, nebo mohou být v diagramu modelovány samostatně na nejvyšší úrovni. Task, někdy také úlohy, jsou atomické aktivity, které se dále nedělí, a tudíž nemohou obsahovat další aktivity. Aby bylo možné tyto dvě aktivity od sebe rozeznat, je pro aktivitu typu sub-proces použit symbol „+“. Pro popis aktivit se používá sloveso v přítomném čase.

Business proces (Byznys proces)

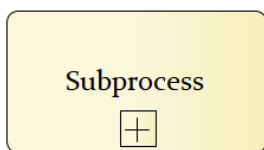
Byznys proces je základní element, který rozděluje celý systém na jednotlivé procesy. Tento proces zajišťuje naplnění podnikových cílů, reaguje na významné události a zajišťuje produkci výstupů (produktů a služeb). Byznys procesy dělíme do tří základních kategorií, kterými jsou hlavní procesy, podpůrné procesy a řídicí procesy. Každý z těchto byznys procesů se pak skládá ze sub-procesů a tasků. Byznys proces je tedy nejvyšším procesem, který napomáhá k rozdělení aktivit do náležitých procesů a vytvoření potřební hierarchie. (OMG, 2014)



Obrázek 15 Business Process (zdroj: [17], vlastní úprava)

Sub-process (Sub-proces)

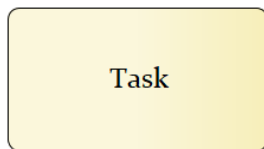
Sub-proces je aktivita, která obsahuje vnitřní modely, které obsahují aktivity, událost a sekvenční toky. V této aktivitě tak vznikají další podrobnější procesy, které modelují dílčí stav celého procesu. Sub-proces může být dvojího typu. První je typu Is Expanded, kdy dílčí proces je přímo viditelný v procesu. Tento model je vhodný jen pro malý počet aktivit. Druhým typem je vnořený diagram, který by se měl používat vždy, pokud bude sub-proces obsahovat větší množství elementů, které by způsobovaly nepřehlednost modelu. (OMG, 2014)



Obrázek 16 Sub-proces (zdroj: [17], vlastní úprava)

Task (Úkol)

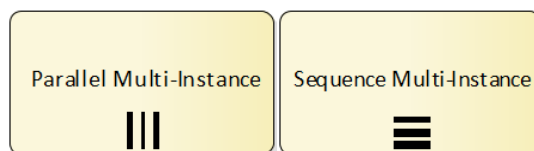
Úkol je atomická aktivita, která je součástí procesu. Jedná se o nejnižší úroveň aktivity, která již nemůže obsahovat dílčí procesy. Při modelování se tvoří všechny aktivity jako tasky a mají-li mít hlubší význam, je nutné je změnit na některý ze sub-procesů nebo aktivit, které budou popsány níže. (OMG, 2014)



Obrázek 17 Úkol (zdroj: [17], vlastní úprava)

Multi-Instance (Vícenásobná instance)

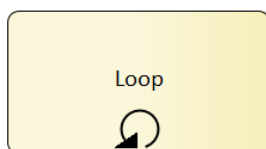
Multi-Instance je typ aktivity, která umožňuje vytvářet mnoho instancí právě v rámci jedné aktivity. Instance mohou být vytvářeny současně (paralelně) např. jeden druh výrobku je vyráběn současně na více strojích. Tato aktivita je označována třemi svislými čarami. Druhým typem multi-instance je sekvenční vykonávání aktivity, to se značí třemi vodorovnými čarami. Příkladem může být tataž výroba určitého výrobku, ale z důvodu nedostatku strojů v danou chvíli může být vyráběn pouze jeden výrobek. Multi-Instance může být modelována jako task nebo jako sub-proces, který hlouběji popisuje danou činnost. (OMG, 2014)



Obrázek 18 Multi-Instance (zdroj: [17], vlastní úprava)

Loop (Opakování)

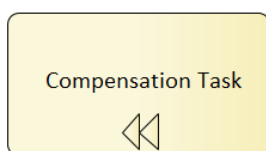
Loop je typ aktivity, která umožňuje opakované provádění určité činnosti založené na Boolean podmínce. To znamená, že aktivita se bude tak dlouho opakovat, dokud nebude podmínka splněna a až poté může tok pokračovat dál. Splnění podmínky se kontroluje po každém kroku. Příkladem může být výroba stejná jako v případě sériového Multi-Instance. Výroba se opakuje do té doby, dokud není vyroben určitý počet výrobků. Ve chvíli, kdy je tento počet splněn, je výroba ukončena a zboží se může odvést. Stejně tak jako u Multi-Instance může být i tato aktivita modelována jako task nebo sub-proces. (OMG, 2014)



Obrázek 19 Loop (zdroj: [17], vlastní úprava)

Compensation (Vrácení do předchozího stavu)

Jedná se o aktivitu, která uvede prostředí do určitého stavu, ve kterém se v dřívější době nacházelo. Příkladem této aktivity může být zrušení rezervace hotelu. Ve chvíli, kdy dojde ke zrušení rezervace, je zabořovaný pokoj v hotelu vrácen do stavu dostupný, aby si jej mohl objednat někdo jiný. Tato aktivita je velmi často spojována se stejnojmennou událostí. Je však nesmírně důležité zdůraznit, že v tomto případě se nepoužívá klasicky vazba Sequence Flow, ale Association. (OMG, 2014)



Obrázek 20 Compensation Task (zdroj: [17], vlastní úprava)

Typy Tasků

V jazyce BPMN existuje mnoho Tasků (úkolů) jak již bylo zmíněno výše. Každý z těchto úkolů však může být zpracováván jiným způsobem, a proto je nezbytné, aby se i toto značení využívalo. Některým uživatelům toto značení přijde zbytečné a myslí si, že je to jen další složitost, která znepráhlední daný model. Opak je ale pravdou. Využívání významu u jednotlivých tasků umožní systémovému návrháři mnohem rychlejší orientaci v procesu, neboť z jednotlivých aktivit je hned patrné, jakým způsobem je vykonávána.

Service Task

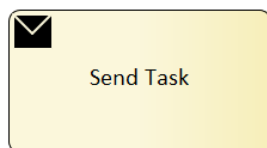
Servisní úloha se používá k znázornění, kdy daná aktivita je využívána určitou službou. Tato služba může být např. webová služba nebo se může jednat o nějakou aplikaci. To znamená, že aktivitu nevytváří aktér a ani do této aktivity nezasahuje, ale aktivita je vytvářena systémem. (OMG, 2014)



Obrázek 21 Service Task (zdroj: [17], vlastní úprava)

Send Task

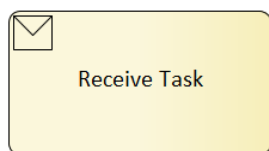
Send Task je aktivitou, která v případě, že do ní vstoupí tok, má za úkol odeslat určitou zprávu. Může odeslat i datový obsah do jiné aktivity nebo účastníkovi. Ve chvíli, kdy je zpráva odeslána, je tato aktivita ukončena a tok pokračuje dál.



Obrázek 22 Send Task (zdroj: [17], vlastní úprava)

Receive Task

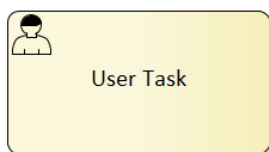
Receive Task je aktivitou, které čeká na obdržení zprávy od externího aktéra. Ve chvíli, kdy je zpráva obdržena se aktivita dokončí a tok pokračuje dál.



Obrázek 23 Receive Task (zdroj: [17], vlastní úprava)

User Task

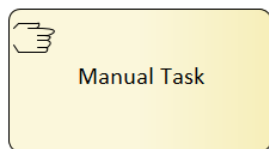
User Task je typická aktivita, ve které pracovník vykonává určitý pracovní postup za pomoci softwarové aplikace. Tento typ aktivity se při modelování vyskytuje nejčastěji.



Obrázek 24 User Task (zdroj: [17], vlastní úprava)

Manual Task

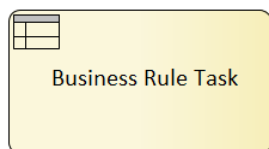
Manual Task je aktivitou, u které se očekává, že jí bude aktér vykonávat manuálně bez jakéhokoli mechanismu nebo aplikace, které by mu pomáhaly při vykonání této činnosti.



Obrázek 25 Manual Task (zdroj: [17], vlastní úprava)

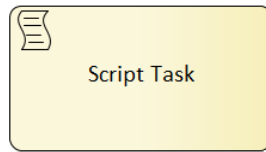
Business Rule Task

Aktivita Business Rule se používá k modelování komplexních rozhodnutí a pravidel. Business rules jsou specifické typy služeb, které spravují byznys aktéři nikoli IT aktéři.



Obrázek 26 Rule Task (zdroj: [17], vlastní úprava)

Script Task Script Task je aktivita, která je vykonávána v určitém jazyce (skriptu), který uživatel předem definuje. Ve chvíli, kdy dojde tok do této aktivity, je daný skript proveden a po jeho dokončení je aktivita ukončena a tok pokračuje dál.



Obrázek 27 Script Task (zdroj: [17], vlastní úprava)

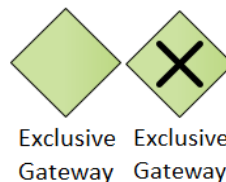
Gateways (Rozhodovací uzly)

Rozhodovací uzly jsou poslední skupinou tokových objektů. Jsou určeny k tomu, aby upřesnily směr, jakým se tok v procesu bude pohybovat mezi aktivitami. Každý z uzlů má své podmínky, které vždy definují, jak bude s tokem dat naloženo.

Exclusive Gateway (XOR)

Tento uzel představuje logický XOR, to znamená, že na vstupu nebo výstupu může nastat právě jedna z možných variant. Každý z uzlů má dvojí použití, a to tehdy, když do něho přichází pouze jedna cesta a z něho vychází více cest. V takovém případě je použití uzlu jednodušší. V případě že z uzlu vystupují pouze dvě cesty, měla by se vždy používat podmínka, na kterou je odpověď ve tvaru ANO/NE. Po splnění jedné z možných variant se touto cestou tok vydá. V případě, že z uzlu vychází více cest, často se jedná i o složitější podmínku, která určí, kterým směrem se tok vydá. V složitých podmínkách se však může stát, že není splněna žádná z podmínek, a proto by měla být jedna z cest označena Sequence Flow typu default. Tato speciální vazba umožní dokončení rozhodovacího toku v případě, že nebyla splněna podmínka. (OMG, 2014)

Druhým typem použití uzlu je, když je v něm slučováno více cest. V takovém případě musí být uzel zase řádně popsán podmínkou, kterou tok musí splnit. Ve chvíli, kdy jeden z toků splní danou podmínku, pokračuje dál a ostatní aktivity zaniknou. Tedy pouze jeden tok může projít tímto uzlem.



Obrázek 28 Exclusive Gateway (zdroj: [17], vlastní úprava)

Inclusive Gateway (OR)

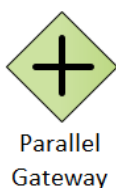
Inclusive Gateway je velmi podobný uzlu XOR s tím rozdílem, že podmínku může splnit jedna nebo všechny příchozí cesty. Podobně i zde v případě složitějších podmínek by měla být stanovena Defaultní cesta, kterou tok může odejít v případě nesplnění podmínky. Pokud je OR uzel použit druhým způsobem, znamená to, že do uzlu přichází více cest a je vždy nutné, aby bylo dopředu známo, kolik cest má být synchronizováno. To bývá splněno vstupní podmínkou (použití OR jak k rozdělení toku, tak k jeho sloužení). (OMG, 2014)



Obrázek 29 Inclusive Gateway (zdroj: [17], vlastní úprava)

Parallel Gateway (AND)

Parallel Gateway je jedním z logicky nejjednodušších uzlů. Tento uzel jako jediný neobsahuje podmínku a tok pokračuje vždy všemi možnými cestami. I v případě druhého způsobu využití tohoto uzlu se postupuje stejným způsobem. Uzel spojí všechny příchozí toky do jednoho, který pak pokračuje dál. Důležitým faktem tohoto uzlu je, že v případě, že některý z toků se zdrží ve svém procesu, tak ostatní toky na tento tok musí čekat a k sloučení dojde až ve chvíli, kdy do uzlu dorazí všechny možné toky. (OMG, 2014)



Obrázek 30 Parallel Gateway (zdroj: [17], vlastní úprava)

Event-Base Gateway

Tento uzel je založen na událostech, které rozhodují, jaký směrem se tok vydá. Za uzlem tedy nejsou aktivity, jak to bývá běžné, ale nacházejí se ve směru toku události. Event-Base je jediným uzlem, který umožňuje, aby z něho tok pokračoval do události. Ve chvíli, kdy tok dorazí do zmíněného uzlu, se zde zastaví a čeká, dokud nebude spuštěna některé z událostí. Ve chvíli spuštění první události se tok

ubírá směrem, na který byla událost spuštěna. Ostatní cesty jsou uzavřeny a již nemůžou být spuštěny, podobně jako v XOR uzlu. (OMG, 2014)



Event-Base
Gateway

Obrázek 31 Event-Base Gateway (zdroj: [17], vlastní úprava)

Parallel Event-Base Gateway

Tento uzel vychází z uzlu Event-Base a stejně tak je v uzlu tok rozdělen a spuštěn u následující události. Rozdíl je v tom, že ostatní cesty ani události nezanikají a v případě, že dojde ke spuštění další události, vzniká nový tok, který pokračuje cestou, ve které byla tato nová událost spuštěna. (OMG, 2014)



Parallel
Event-Base
Gateway

Obrázek 32 Parallel Event-Base Gateway (zdroj: [17], vlastní úprava)

Complex Gateway

Complex Gateway je jedním z nejsložitějších uzlů, proto je doporučeno ho používat jen v případech, kdy není možné použít jiný uzel. Tento uzel je podobný jako Inclusive Gateway s tím rozdílem, že obsahuje složitější podmínku, která může být popsána výpočtem nebo textovým popisem. V některých případech je tento uzel nutné použít, a proto zde bude uveden jednoduchý příklad, jak může být tento uzel aplikován: $x_1 + x_2 + \dots + x_m \geq 3$. Tento příklad říká, že uzel bude čekat do té doby, než do něho přitečou 1 až 3 toky ve chvíli, kdy je tato podmínka splněna. Tok pak může pokračovat dál. Jako další příklad může být uvedena podmínka 2 ze 3. To znamená, že do uzlu musí dorazit přesně dva toky, aby mohly být sloučeny a pokračovat dál. (OMG, 2014)



Complex
Gateway

Obrázek 33 Complex Gateway (zdroj: [17], vlastní úprava)

3.3.2.2 Connecting Objects (Spojovací objekty)

Spojovací objekty slouží pro znázornění posloupnosti elementů v procesu. Elementy toků se pomocí vazeb odkazují na ostatní elementy, které jsou s těmito elementy navzájem propojeny tak, aby vytvořily úplný tok procesu.

Sequence Flow

Sekvenční tok vyjadřuje posloupnost elementů v procesu. Každý sekvenční tok může mít pouze jeden zdrojový a jeden cílový element. Může spojovat pouze tokové elementy (flow element). Příkladem sekvenční vazby může být spojení mezi událostí a aktivitou.



Obrázek 34 Sequence Flow (zdroj: [17], vlastní úprava)

Conditional Sequence Flow

Sekvenční vazba může obsahovat podmínku a to znamená, že tok půjde po této vazbě až ve chvíli, kdy bude splněna podmínka. Tato podmínka u vazby funguje stejným způsobem jako podmínka v uzlu s tím rozdílem, že aktivita, ze které vychází, může obsahovat i obyčejnou sekvenční vazbu. Touto vazbou půjde tok vždy, zatímco sekvenční vazbou s podmínkou může projít tok, jen pokud je splněna podmínka.



Obrázek 35 Conditional Sequence Flow (zdroj: [17], vlastní úprava)

Default Sequence Flow

Jedná se o vazbu, která může nastat pouze ve chvíli, kdy jiná cesta není možná. Tato vazba již byla zmíněna v rámci popisu uzlů (gateway). V případě, že z aktivity nebo z uzlu vede více vazeb, může nastat situace, kdy tok nebude moci z nějakého důvodu projít ani jednou z nich, tedy je nutné použít i tuto vazbu, aby nedošlo k uvěznění toku. Defaultní tok se může používat i ve spojení se sekvenčním tokem

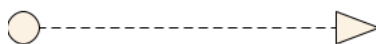
a podmínkovým sekvenčním tokem. Např. v případě, kdy není možné použít gateway, je možné jej nahradit pomocí podmínkového sekvenčního toku a defaultní vazby.



Obrázek 36 Default Sequence Flow (zdroj: [17], vlastní úprava)

Message Flow

Tato vazba se používá k zobrazení toku zpráv mezi dvěma aktéry. Nejčastěji se tato vazba používá k propojení dvou poolů, které reprezentují aktéra nebo systém. Message Flow může být také napojena na událost, která pak může spouštět další proces.



Obrázek 37 Message Flow (zdroj: [17], vlastní úprava)

Association

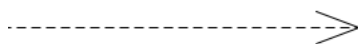
Asociace se nejčastěji používá pro spojení elementů s artefakty. Dále je možné vazbu využívat ke spojení dvou elementů, které nemají jasně specifikovaný vztah. Pro označení směru toku informací se k vazbě přidává šipka.



Obrázek 38 Association (zdroj: [17], vlastní úprava)

Data Association

Data Associations se používají ke spojení elementů s datovými objekty nebo datovými úložišti tak, aby znázornila, jak dané elementy s daty pracují.



Obrázek 39 Data Association (zdroj: [17], vlastní úprava)

3.3.2.3 Swimlanes

Plavecké dráhy (Swimlanes) zahrnují jak bazény (Pools), tak jednotlivé dráhy (Lanes). Tyto elementy se využívají především k upřesnění a zpřehlednění diagramů. Ve chvíli, kdy je potřeba znázornit komunikaci více účastníků (mohou to být aktéři nebo software), využívají se pro jejich znázornění právě tyto elementy. (OMG, 2014)

Pools (Bazén)

Tyto elementy jsou určeny k tomu, aby umožnily v procesu oddělení jednotlivých aktérů. Každý diagram by měl obsahovat pouze jeden proces. Někdy je však zapotřebí znázornit komunikaci mezi více aktéry (procesy), kteří spolu potřebují komunikovat pro výkon své aktivity. Příkladem mohou být dva systémy, které mezi sebou komunikují a předávají si určitá data. Každý z těchto systémů bude mít svůj bazén, ve kterém bude mít aktivity a svůj tok. Pro komunikaci mezi těmito bazény se používá vazba Message flow. Je to jediná vazba, která může být mezi bazény použita. Komunikace mezi bazény může být dvojího typu. První možností je, že jsou viditelné všechny procesy, tedy je znázorněná celá komunikace jak mezi bazény, tak i tok v procesech. Druhou možností je, že procesy jednoho z bazénů nejsou pro čtenáře k dispozici a je tak možné vidět pouze procesy jednoho bazénu (systému) a s druhým je znázorněna pouze komunikace. Nicméně vnitřní procesy známy nejsou. (OMG, 2014)



Obrázek 40 Pool (zdroj: [17], vlastní úprava)

Lanes (Dráhy)

Dráhy, někdy také označovány jako plavecké dráhy, mohou být modelovány pouze jako součást bazénu (Pool). Slouží k jeho dalšímu upřesnění a rozdělení. Příkladem může být systém obsahující více podsystémů nebo obsahující různé moduly, které jsou součástí tohoto systému, ale mají své aktivity. Tyto aktivity napomáhají k zhotovení celého procesu. Na rozdíl od komunikace mezi bazény, dráhy v bazénu se chovají jako jeden bazén (proces) a tudíž komunikace nemusí probíhat pomocí message flow, ale data se mohou přenášet tokem pomocí vazby sequence flow jako v klasickém procesu. (OMG, 2014)



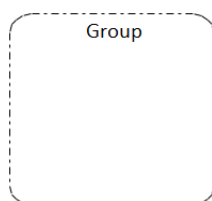
Obrázek 41 Lane (zdroj: [17], vlastní úprava)

3.3.2.4 Artifacts

Artefakty umožňují vizuální reprezentaci, která dovolí přidat popis k elementům mimo hlavní proces. Artefakty mohou představovat data nebo poznámky popisující a zpřesňující aktivity. (OMG, 2014)

Group (Skupina)

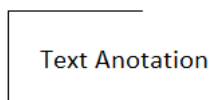
Skupina se používá pro účely dokumentace nebo zpřehlednění modelu, aby seskupila aktivity, které mají společné charakteristiky. Tento element neovlivňuje přímo sekvenční tok, slouží pouze jako reprezentace a nemůže tedy být ani nahrazen za element Pool. (OMG, 2014)



Obrázek 42 Group (zdroj: [17], vlastní úprava)

Text Annotation (Anotace)

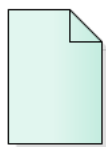
Anotace umožňuje čtenáři poskytnout krátké doplňující informace ohledně elementů nebo celého diagramu. (OMG, 2014)



Obrázek 43 Text Annotation (zdroj: [17], vlastní úprava)

Data Objects

Datový objekt reprezentuje určitá data, která jsou vytvářena v určité aktivitě nebo jsou aktivitou vyžadována, aby tato aktivita mohla být provedena. Datové objekty jsou s aktivitami propojovány pomocí asociační vazby. Příkladem datového objektu mohou být plány vstupující do výroby. Kdyby dotýčný pracovník neměl potřebné nákresy, nemohl by danou součástku vyrobit. Datové objekty je možné označovat šipkou na elementu dle toho, zda se jedná o data vstupující do aktivity nebo z ní vystupující. Tento způsob se však nedoporučuje z důvodu znovupoužitelnosti daného elementu (Datový objekt může být výstupem z jedné aktivity a ten samý datový objekt může vstupovat do jiné aktivity). Pro znázornění směru dat je vhodnější používat šipku u vazby. (OMG, 2014)

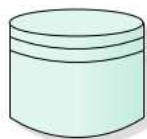


Data
Object

Obrázek 44 Datový objekt (zdroj: [17], vlastní úprava)

Data Stores

Datové úložiště umožňuje aktivitám používat nebo upravovat data, která se nenacházejí přímo v daném procesu.



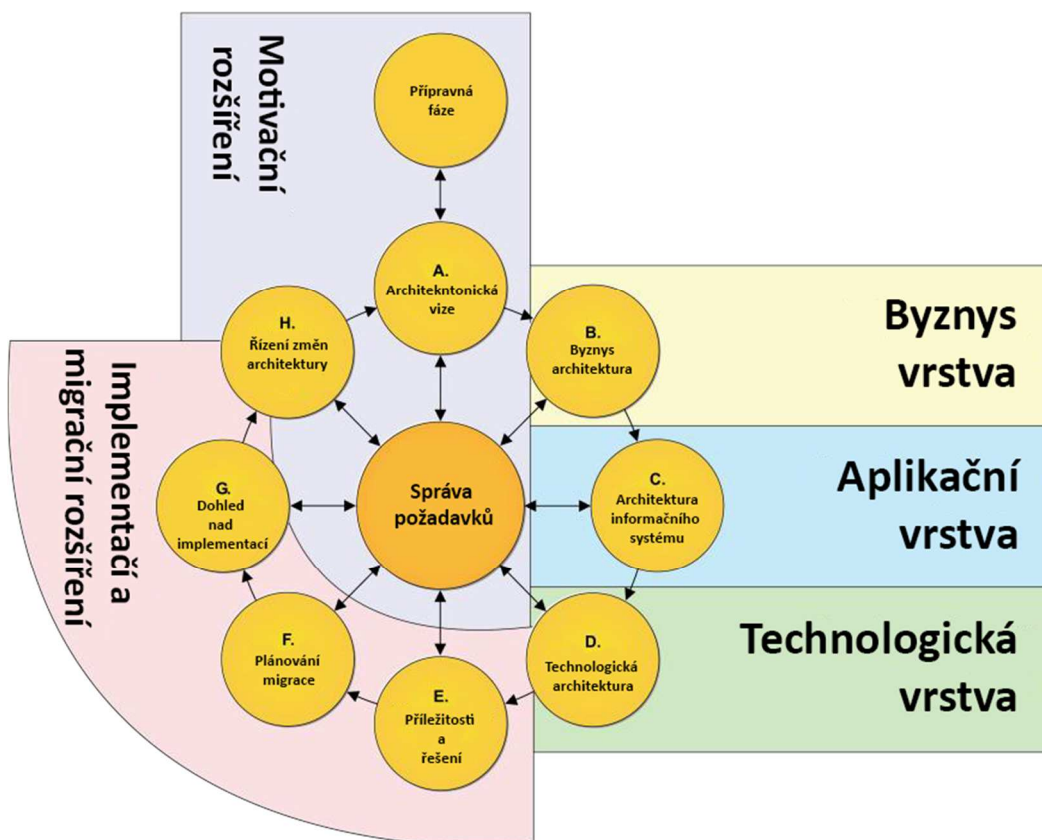
Data
Store

Obrázek 45 Data Store (zdroj: [17], vlastní úprava)

3.4 ArchiMate

ArchiMate je otevřený modelovací jazyk sloužící pro grafické znázornění podnikové architektury, který je v současné době spravovaný společností The Open Group. V současné době se používá nejnovější verze 3.0.1, která byla vydána v červnu roku 2016 a je značně rozšířena a vylepšena oproti původní verzi 2.1. Jazyk ArchiMate slouží k vytvoření uceleného pohledu na společnost, není však určen pro vytváření příliš detailních modelů. Proto je nutné zvolit správnou míru podrobnosti a tím i odpovídající jazyk. Žádný z jazyků by však neměl překračovat svůj účel a určitou míru detailnosti, neboť by se popis stal příliš podrobným a nemohl by splnit svůj záměr, ke kterému byl navrhnut. Jak již bylo zmíněno jazyk ArchiMate je možné propojit s jinými jazyky, jako je UML nebo BPMN, které mohou vybrané oblasti zmodelovat podrobněji. ArchiMate i BPMN využívají pro modelování byznys procesy, rozdíl je však v jejich aplikaci. ArchiMate modeluje procesy na abstraktní úrovni, které jsou nezbytné pro návrh podnikové architektury, ale nejsou určeny pro detailní modelování procesů, zatímco BPMN je cíleno k modelování detailnějších procesů zahrnujících atomické tasky. K jazyku UML má ArchiMate ještě blíže, jelikož většina elementů a vazeb z tohoto jazyka vychází.

ArchiMate sám o sobě není metodikou, vychází však z metodiky TOGAF, která obsahuje metody a nástroje pro tvorbu, údržbu a řízení podnikové architektury. TOGAF klade důraz na vytvoření jednotlivých kroků architektury, které budou mít za cíl konstruovat jednotné modely upravené na míru organizacím. Struktura jazyka ArchiMate se skládá ze 3 hlavních částí (byznys vrstva, aplikační vrstva a technologické vrstvy), které jsou doplněny o motivační rozšíření, implementační a migrační rozšíření kompletující celý rámec TOGAF. Obrázek níže znázorňuje propojení jazyka ArchiMate a metodiky TOGAF. (THE OPEN GROUP, c2012-2017; THE OPEN GROUP, c1995-2018)



Obrázek 46 Vztah mezi jazykem ArchiMate a metodikou TOGAF (zdroj: [24], vlastní úprava)

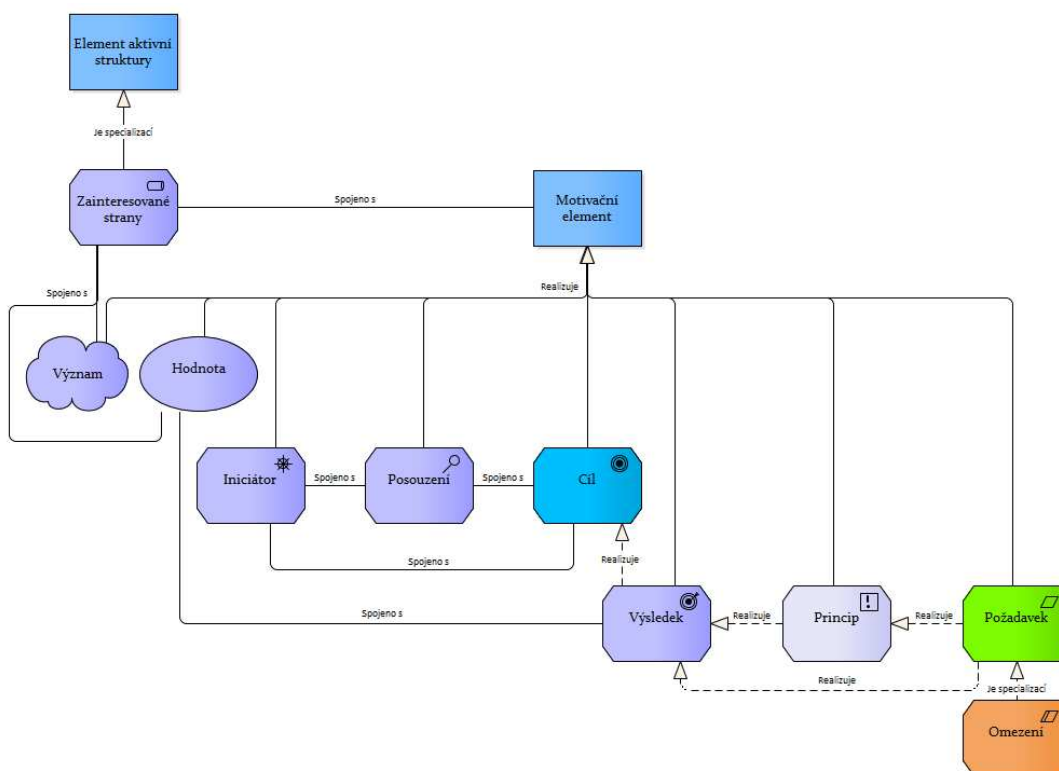
3.4.1 Motivation Elements (Motivační elementy)

Motivační elementy se používají k modelování motivačních faktorů, které vedou k návrhu, nebo změnám podnikové architektury. Motivační elementy se rozdělují do tří skupin. V první skupině se nacházejí elementy jako iniciátor (driver), který

může pocházet z modelované společnosti nebo může stát mimo daný podnik. Dalším elementem je zainteresovaná strana (stakeholder), která může reprezentovat jedince nebo skupinu lidí, jako je projektový tým, podnik nebo společnost. A posledním elementem první skupiny je posouzení (assessment), které slouží k nalezení potřebných faktorů ovlivňujících firmu, ze kterých se posléze dá vytvořit SWOT analýza.

Do druhé skupiny patří cíl, výsledek, princip, požadavek a omezení. Cíl je jedním z nejdůležitějších elementů organizační struktury, neboť se jedná o určitou vizi, která motivuje společnost jít dál. Při modelování diagramu pomocí motivačních elementů se také někdy tento diagram může označovat jako cíle, neboť cíl znázorňuje, že účastník chce dosáhnout určitého výsledku (např. zvýšení spokojenosti zákazníků). Principy a požadavky představují požadované vlastnosti nebo postupy řešení, které vedou realizaci stanovených cílů.

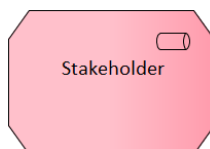
Do poslední skupiny patří dva elementy. Význam (meaning), který napomáhá ujasnit význam nebo interpretaci hlavních elementů architektury. Druhým elementem je hodnota (value). Ta umožňuje výsledku nebo jiným elementům přiřadit jejich hodnotu, a to nejčastěji finanční.



Obrázek 47 Metamodel motivačních elementů (zdroj: [24], vlastní úprava)

Stakeholder (Zainteresované strany)

Stakeholder je role reprezentující jednotlivce, tým nebo organizaci, která zastupuje jejich zájmy a ovlivňuje chod podniku. Stakeholder může být definován jako je např. generální ředitel, správní rada, akcionář, zákazník, byznys nebo aplikační architekt nebo také zákonodárny orgán. Pro popis elementu se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 48 Zainteresované strany (zdroj: [24], vlastní úprava)

Driver (Iniciátor)

Iniciátor představuje roli, která motivuje organizaci, aby definovala své cíle a provedla změny potřebné k jejich dosažení. Iniciátor může být interní, v takovém případě bývá obvykle spojován se stakeholderem a často definován jako „concerns“ (tedy nějaký znepokojující faktor). Znepokojení se mohou týkat jakéhokoli aspektu funkce, vývoje nebo fungování systému včetně úvah jako jsou optimální výkonnost, spolehlivost, bezpečnost, distribuce nebo vývoj. Příkladem interních iniciátorů může být spokojenost nebo ziskovost zákazníků. Extrémními iniciátory pak mohou být např. ekonomické změny nebo měnící se právní předpisy. Pro popis elementu se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)

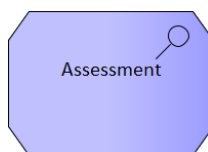


Obrázek 49 Iniciátor (zdroj: [24], vlastní úprava)

Assessment (Posouzení)

Posouzení slouží k odhalení silných, slabých stránek firmy nebo k nalezení hrozeb a příležitostí pro určitou oblast zájmu. Tedy elementy posouzení napomáhají k sestavení SWOT analýzy a následné úpravě stávajících nebo tvorbě nových cílů společnosti. Pro znázornění zde bude použit příklad slabé stránky „zákazníci si stěžují na nefunkční helpdesk“. Po nalezení této slabiny může být definován nový

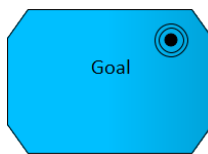
cíl: Zlepšit helpdesk. Pro popis elementu se používá podstatné jméno nebo velmi krátká věta. (THE OPEN GROUP, c2012-2017)



Obrázek 50 Posouzení (zdroj: [24], vlastní úprava)

Goal (Cíl)

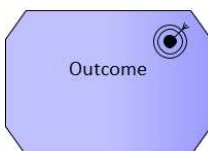
Cíl definuje záměr nebo směr, kterým se společnost a zainteresované strany budou ubírat. V zásadě cíl může reprezentovat cokoli, co si zainteresované strany přejí jako např. zvýšení zisků, snížení čekací doby na helpdesku nebo zavedení online správy portfolia. Cíle se obvykle vyjadřují např. pomocí slov „zvýšení, zlepšení nebo usnadnění“. Dále se cíle mohou dělit na další podcíle, např. zvýšení zisku se rozdělí na podcíle snížení nákladů a zvýšení tržeb. (THE OPEN GROUP, c2012-2017)



Obrázek 51 Cíl (zdroj: [24], vlastní úprava)

Outcome (Výsledek)

Výsledek představuje konečný stav, kterého bylo dosaženo. Outcome jsou podnikatelsky orientované výsledky na vysoké úrovni, produkováné schopnostmi organizace a také klíčovými elementy podnikavé architektury, které tyto schopnosti realizují. Výsledky jsou hmatatelné, kvantitativní, časově závislé a mohou být spojeny s hodnotou (value). Outcome jsou konečné výsledky a cíle. Po požadavky jsou často formulovány na základě těchto výsledků, které by měly být realizovány (např. klíčové dodavatelské partnerství nebo čtvrtletní zisky roku 2017 vzrostly meziročně o 15 %). (THE OPEN GROUP, c2012-2017)



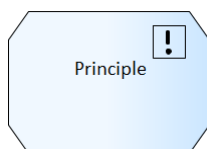
Obrázek 52 Výsledek (zdroj: [24], vlastní úprava)

Principle (Princip)

Princip je definován jako obecný požadavek, který musí být splněn.

Princip silně ovlivňuje cíle a společně s požadavky definuje zamýšlené vlastnosti systému. Na rozdíl od požadavků je princip obecný a má delší časový charakter. Princip definuje obecnou vlastnost, která platí pro jakýkoliv systém podobného charakteru, kdežto požadavek definuje vlastnost, která se vztahuje na konkrétní systém popsany architekturou. (THE OPEN GROUP, c2012-2017)

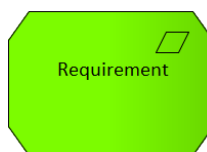
Například princip „procesní řízení informací je v souladu se všemi předpisy“ je realizován požadavky, které jsou stanoveny platnými předpisy, které se vztahují na konkrétní systém, který je ve fázi návrhu.



Obrázek 53 Princip (zdroj: [24], vlastní úprava)

Requirement (Požadavek)

Požadavky modelují vlastnosti, které jsou potřebné k dosažení požadovaného konce, jenž je znázorněn pomocí cíle. Pro příklad lze uvést dva požadavky pro realizaci cíle, který má zlepšit správu portfolia (přiřazení osobního asistenta každému zákazníkovi nebo zavedení online správy portfolia). První z požadavků může být realizován lidským činitelem, kdežto druhý požadavek může být vyřešen pomocí softwarové aplikace. Jednotlivé požadavky je možné dělit na dílčí, podrobnější požadavky. (THE OPEN GROUP, c2012-2017)



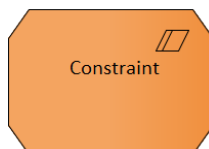
Obrázek 54 Požadavek (zdroj: [24], vlastní úprava)

Constraint (Omezení)

Omezení představuje vlastnost, která brání realizaci cílů.

Na rozdíl od požadavků nepředstavuje omezení určitou funkcionalitu systému, která se má uskutečnit, ale představuje rizika, jež by mohla omezit způsob fungování nebo realizaci. Omezením může být např. specifická technologie, která

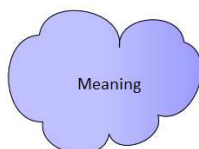
musí být použita nebo čas a rozpočet, jenž by mohl omezit implementaci systému. Jako další příklad může být uvedeno právní omezení, které může limitovat fungování systému. (THE OPEN GROUP, c2012-2017)



Obrázek 55 Omezení (zdroj: [24], vlastní úprava)

Meaning (Význam)

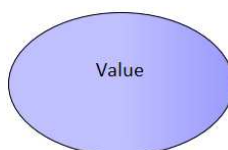
Význam představuje informační záměr používaný k popisu významu prvků pasivní struktury (např. dokumentu, zprávy). Je možné, že různí uživatelé mohou pohlížet na informativní funkce elementu odlišně (např. co bere zákazník jako potvrzení registrace, může společnost vnímat jako změnu zákazníka). Také je možné, aby různé reprezentace měly stejný význam (např. online dokument, vytištěný formulář atd. jsou různé typy dokumentů, které však mohou obsahovat stejné informace). Pro popis elementu se používá podstatné jméno nebo krátká věta. (THE OPEN GROUP, c2012-2017)



Obrázek 56 Význam (zdroj: [24], vlastní úprava)

Value (Hodnota)

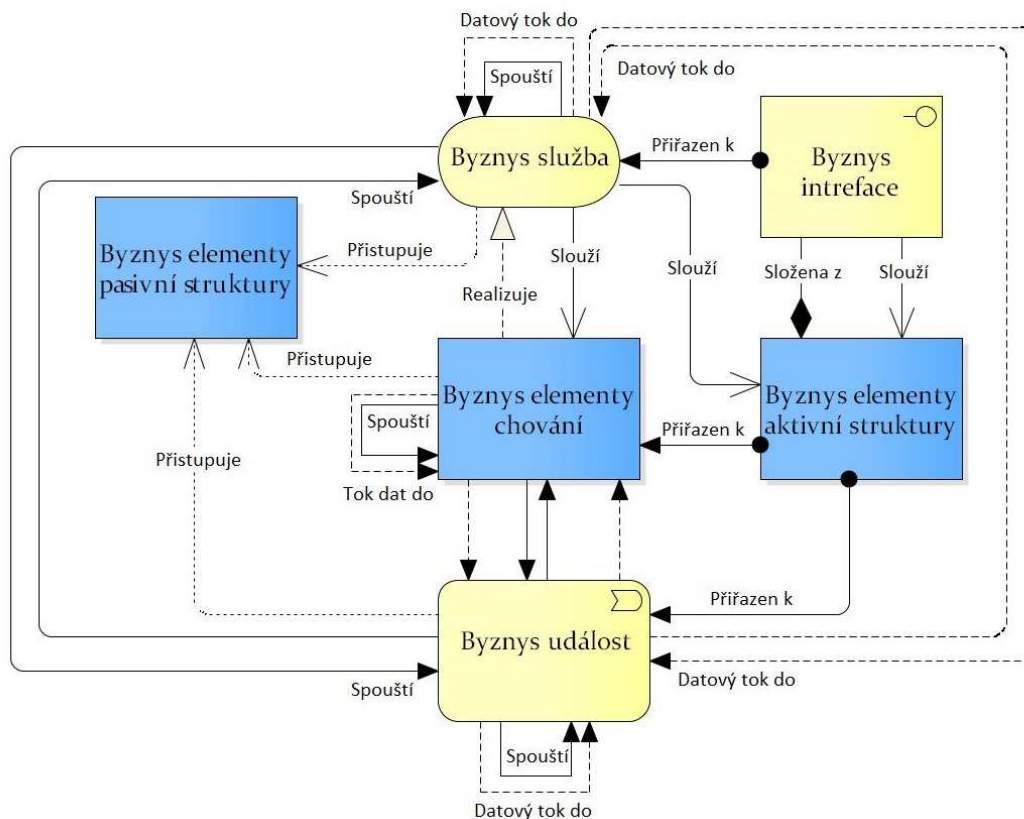
Hodnota reprezentuje relativní hodnotu, cenu, užitek nebo důležitost základních elementů (elementy struktury a chování) či výsledků (outcome). Hodnota je často vyjádřena v penězích, avšak nefinanční hodnota je pro podnik také velmi důležitá (např. praktická, funkční, informační nebo znalostní hodnota). Pro popis elementu hodnoty by mělo být použito vyjádření užitku z daného produktu nebo služby, který přinese svému uživateli. (THE OPEN GROUP, c2012-2017)



Obrázek 57 Hodnota (zdroj: [24], vlastní úprava)

Business Layer (Byznys vrstva)

Obrázek níže popisuje přehled elementů byznys vrstvy společně s vazbami mezi nimi. Byznys elementy chování, elementy aktivní struktury a elementy pasivní struktury jsou pouze abstraktními elementy a jejich specializace je popsána v následující části. Model nezobrazuje všechny povolené vazby, slouží pouze pro jejich znázornění. (THE OPEN GROUP, c2012-2017)



Obrázek 58 Metamodel byznys vrstvy (zdroj: [24], vlastní úprava)

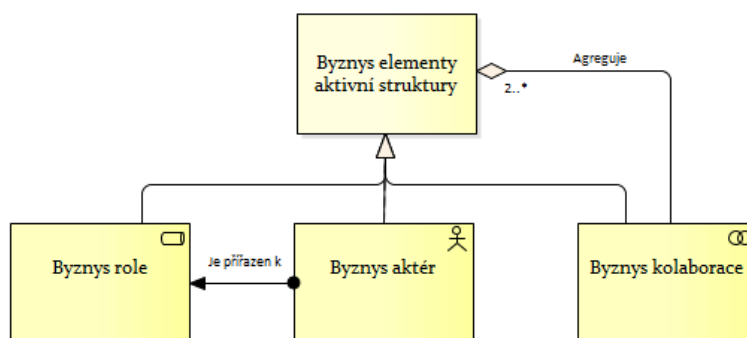
3.4.1.1 Active Structure Elements (Elementy aktivní struktury)

Elementy aktivní struktury byznys vrstvy znázorňují statistickou strukturu organizace. Mezi aktivní elementy patří subjekty (např. byznys aktér nebo byznys role), které jsou schopny vykonávat určité aktivity (např. byznys procesy nebo byznys funkce).

Byznys aktér (business actor) může reprezentovat jednotlivce (např. zákazníka nebo zaměstnance), ale také skupiny, oddělení nebo celé organizace. Popisy architektury se zaměřují na strukturu, což znamená, že vzájemné vztahy jednotlivých elementů při modelování organizace hrají důležitou roli.

Aby bylo možné spojit jednotlivé prvky, které vykonávají kolektivní aktivitu, vznikla byznys součinnost (business collaboration), která spojuje více byznys rolí. Byznys rozhraní (business interface) je jakýmsi přístupovým bodem, který umožní zpřístupnit byznys službu svému okolí. Jedna a tatáž služba může být nabízena různým rozhraním (např. telefon, email nebo internet). Ve většině modelovacích jazyků není obvyklé používat byznys rozhraní s výjimkou aplikačního modelování, kde se tento koncept běžně používá).

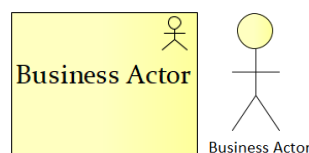
Byznys vrstva využívá tři typy aktivních elementů, které jsou důležité pro její strukturu. Těmito elementy jsou: byznys aktér, byznys role a byznys součinnost. Dále bude využíván element rozhraní a element lokace. Ten bude popsán níže, neboť je přidán ze starší verze jazyka ArchiMate.



Obrázek 59 Byznys elementy aktivní struktury (zdroj: [24], vlastní úprava)

Business Actor (Byznys aktér)

Byznys aktér je definován jako podnikatelská entita, která je schopna vykonávat určité aktivity, přiřazené k jedné nebo více byznys rolím. Byznys aktér může reprezentovat jednotlivce ve firmě, části firmy jako jsou jednotlivá oddělení nebo také celou organizaci, ale i účastníky mimo samotný podnik, kterými může být zákazník nebo partner. (THE OPEN GROUP, c2012-2017)

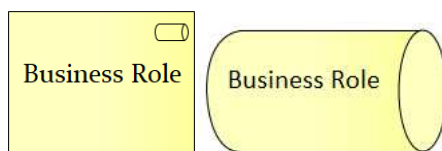


Obrázek 60 Byznys aktér (zdroj: [24], vlastní úprava)

Business Role (Byznys role)

Byznys role je zodpovědná za provedení konkrétní aktivity, ke které může být přiřazen byznys aktér. Byznys role vyžaduje určitou zodpovědnost a znalost

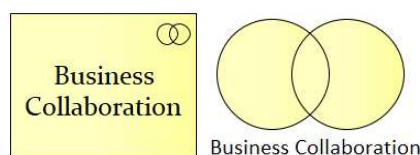
byznys aktéra při vykonávání jeho aktivity nebo funkce. Byznys role také může znázorňovat delegování práce v rámci organizační struktury podniku. K jedné roli tak může být přiřazeno více byznys aktérů. K popisu byznys role se využívá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 61 Byznys role (zdroj: [24], vlastní úprava)

Business Collaboration (Byznys součinnost)

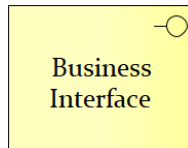
Byznys součinnost je spojením dvou a více byznys rolí, které spolupracují na vykonání určité kolektivní aktivity. Pro popis vnitřního chování jednotlivých byznys rolí jsou používány byznys interakce. Ty popisují interní aktivity vykonávané v rámci jedné součinnosti. Dalo by se tak říct, že element součinnosti obsahuje interakce, které budou více popsány níže. Na rozdíl od rolí, které tato součinnost seskupuje, nemá byznys součinnost ve firmě permanentní účast (bývá časově omezená) a její zaměření je specifické podle rolí, které spojuje. Byznys součinnost je vhodné používat například při modelování B2B interakcí mezi různými organizacemi. Pro popis byznys součinnosti se využívá podstatné jméno, ale je také běžné nechat tento element bez názvu. [24]



Obrázek 62 Byznys součinnost (zdroj: [24], vlastní úprava)

Business Interface (Byznys rozhraní)

Byznys rozhraní je jakýmsi přístupovým bodem, který zpřístupňuje byznys službu svému prostředí. Byznys rozhraní poskytuje funkcionalitu byznys služby ostatním byznys rolím nebo tuto funkcionalitu od byznys služeb vyžaduje. Velmi často se však nepoužívá obecný název byznys rozhraní, ale je nahrazováno již konkrétními kanály (telefon, internet nebo nějaké konkrétní místo jako sídlo firmy). Z toho vyplývá, že jedna byznys služba může být poskytována prostřednictvím několika různých rozhraní. Pro popis byznys rozhraní se využívá podstatné jméno. (THE OPEN GROUP, c2012-2017)



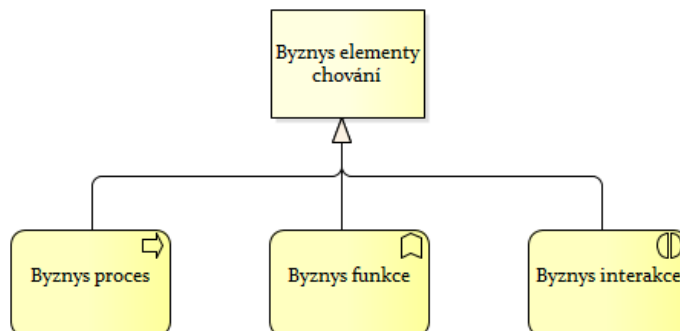
Obrázek 63 Byznys interakce (zdroj: [24], vlastní úprava)

Behavior Elements (Elementy chování)

Na základě orientace na služby, čímž je specifický jazyk ArchiMate, jsou elementy chování rozděleny na interní a externí chování organizace. Externě viditelné aktivity jsou modelovány pomocí byznys služeb, které poskytují svou funkcionalitu okolnímu prostředí tak, aniž by odhalily svou interní funkcionalitu. Na rozdíl od toho interní služby jsou uživateli neviditelné a místo toho slouží k podpoře ostatních procesů nebo funkcí v organizaci. V jazyce ArchiMate existuje několik typů elementů chování, které mohou realizovat služby. V tomto jazyce se rozlišuje mezi procesním a funkčním řízením. Jelikož není vždy úplně možné rozlišit tyto dva způsoby, definuje ArchiMate dva elementy spojené s tímto řízením, kterými jsou byznys proces a byznys funkce. Každý z těchto elementů lze použít pro seskupování více detailních byznys procesů nebo funkcí, ale každý z nich pro tento účel používá jiná skupinová kritéria. Byznys proces představuje tok aktivit (workflow), nebo tok hodnot sestávající se z dílčích procesů nebo funkcí, s jednou nebo více startovacími událostmi vedoucími k určitému výstupu nebo výsledku. Cílem byznys procesu je naplnit potřeby zákazníka, který může být interní nebo externí. Funkce seskupuje aktivity na základě požadovaných dovedností zdrojů, aplikační podpory a dalších faktorů. Typicky jsou byznys procesy definovány na základě produktů a služeb, které organizace nabízí. Zatímco byznys funkce se zabývají např. přidělováním prostředků k úlohám a podporu aplikací. (THE OPEN GROUP, c2012-2017)

Byznys interakce popisuje stejně jako proces nebo funkce jednotku aktivity, ale na rozdíl od zmíněných dvou elementů provádí spolupráci dvou nebo více rolí v organizaci. Interakce se může rozpadat na dílčí interakce. Z pohledu rolí účastnících se spolupráce jsou interakce brány jako externí aktivity. Pro spolupráci jako celek jsou však aktivity brány jako interní. Podobně jako procesy nebo funkce může být byznys interakce zpřístupněna okolí prostřednictvím byznys služeb.

Byznys událost je určitý typ akce, která může mít vliv na byznys procesy, funkce nebo interakce (např. je spustit nebo přerušit). Tato událost je velmi podobná události z jazyka BPMN.

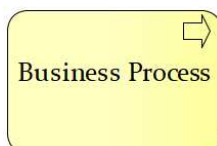


Obrázek 64 Byznys elementy chování (zdroj: [24], vlastní úprava)

Business process (Byznys proces)

Byznys proces představuje posloupnost podnikových aktivit na základě pořadí, ve kterém se ve firmě vyskytují. Byznys proces popisuje interní aktivity, které jsou vykonávané byznys rolí a jejímž výstupem jsou produkty nebo služby. Pro zákazníka jsou relevantní pouze konečné výrobky a služby. Byznys procesy firmy zákazníka nezajímají, představují pro něho pouze černou skříňku, která skrývá vnitřní aktivity společnosti. Z tohoto důvodu se byznys procesy označují jako interní. Složitý byznys proces se může rozpadat na jednotlivé sub-procesy a tasky, které více konkretizují jednotlivé aktivity ve firmě. Tyto aktivity na nižší úrovni mohou být spojeny se specifickou rolí, která je agregována rolí původní. Pro popis názvu byznys procesu se používá sloveso v přítomném čase (např. zahájit výrobu, naskladnit zboží, odeslat fakturu). (THE OPEN GROUP, c2012-2017)

Modelovací jazyk ArchiMate je určen pro znázornění existence procesů a nikoli pro jejich detailní popis. Na modelování těchto detailních procesů je vhodné využít sofistikovanější jazyk pro to učení, kterým je BPMN.



Obrázek 65 Byznys proces (zdroj: [24], vlastní úprava)

Business function (Byznys funkce)

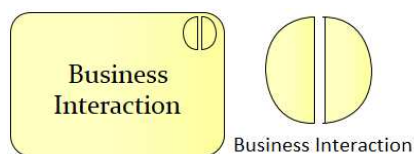
Byznys funkce představuje element chování, který sdružuje aktivity na základě společných kritérií (například podle požadovaných zdrojů nebo kompetencí), které jsou úzce spjaty s organizací, ale nemusí být nutně touto organizací řízeny. Stejně jako byznys proces i byznys funkce popisují aktivity podniku, ale liší se ve způsobu, jakým jsou seskupovány. Byznys procesy jsou seskupovány na základě posloupnosti aktivit nebo jejich toků, které jsou potřebné k realizaci celého procesu. Kdežto u byznys funkcí jsou elementy seskupovány dle požadovaných podnikových zdrojů, dovedností, kompetencí, znalostí atd. Popis názvu byznys funkce by měl jasně odrážet význam funkce, proto se pro popis používá slovesné podstatné jméno (např. recyklace, zpracování, uklízení). (THE OPEN GROUP, c2012-2017)



Obrázek 66 Byznys funkce (zdroj: [24], vlastní úprava)

Business Interaction (Byznys interakce)

Byznys interakce je element kolektivního chování spolupracující na dvou nebo více byznys rolích. Interakce je podobná byznys procesu nebo funkci, ale s tím rozdílem, že je realizována spoluprací více rolí, které sdílejí odpovědnost za jejich vykonání. Pro popis byznys interakce by se mělo využívat sloveso v přítomném čase. (THE OPEN GROUP, c2012-2017)

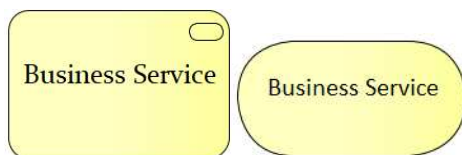


Obrázek 67 Byznys interakce (zdroj: [24], vlastní úprava)

Business Service (Byznys služba)

Byznys služba představuje jednoznačný popis činností dané aktivity. Byznys služba poskytuje funkcionalitu byznys role nebo spolupráce svému prostředí, a to prostřednictvím jednoho nebo více byznys rozhraní. Byznys služba nabízí jednotku chování, která má z pohledu prostředí smysl (význam) a užitek. Byznys služby mohou být externí zákaznické služby (např. cestovní pojištění) nebo interní

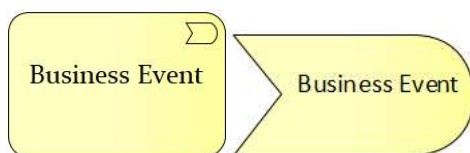
podpůrné služby (např. správa firemních zdrojů). V názvu může být také explicitně obsaženo slovo „služba“ (např. služba analýza procesů). (THE OPEN GROUP, c2012-2017)



Obrázek 68 Byznys služba (zdroj: [24], vlastní úprava)

Business Event (Byznys událost)

Byznys událost představuje element ovlivňující byznys aktivity. Byznys procesy a další elementy chování byznys vrstvy mohou být událostí spuštěny nebo přerušeny. Byznys procesy mohou vyvolat události, které spustí další byznys procesy, funkce nebo interakce. Na rozdíl od procesů, funkcí a interakcí je však byznys událost okamžitá a nemá dobu trvání. Byznys události jsou podobné událostem jazyka BPMN (např. se může jednat o událost, která spouští opakující se byznys proces, který je proveden každé první pondělí v měsíci). Pro popis názvu byznys události se používá sloveso v minulém čase (např. pohledávka přijata). (THE OPEN GROUP, c2012-2017)



Obrázek 69 Byznys událost (zdroj: [24], vlastní úprava)

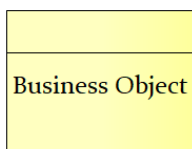
3.4.1.2 Passive Structure Elements (Elementy pasivní struktury)

Pasivní struktura byznys vrstvy obsahuje elementy pasivní struktury, se kterými manipulují elementy chování, jako jsou byznys procesy nebo funkce. V byznys vrstvě existují dva hlavní typy pasivních elementů, kterými jsou byznys objekt a reprezentace. Dále se zde nachází smlouva (contract), která se nejčastěji používá společně s výrobkem, jenž je specifikací byznys objektu. (THE OPEN GROUP, c2012-2017)

Business Object (Byznys objekt)

Byznys objekt je definován jako pasivní element, který je významný z hlediska podnikání. Jazyk ArchiMate se obecně zaměřuje na modelování typu objektů a nikoli konkrétních instancí. Proto se byznys objekt ve většině případů modeluje

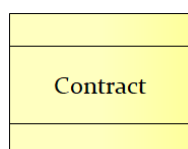
jako typ objektu (stejně jako třída v jazyce UML), z nichž několik se může nacházet v rámci společnosti. Byznys objekty jsou pasivní v tom smyslu, že nespouští ani nevykonávají jiné procesy. Byznys objekt je používán k reprezentaci důležitých informačních aktiv, které podnik vytváří a používá během svého podnikání (smlouvy, faktury, data o zákaznících). Pro popis názvu byznys objektu se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 70 Byznys objekt (zdroj: [24], vlastní úprava)

Contract (Kontrakt)

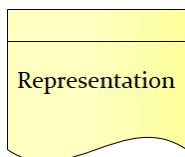
Kontrakt je definován jako element představující formální nebo neformální specifikaci dohody (smlouvy) mezi poskytovatelem a spotřebitelem, která upravuje práva a povinnosti spojené s produktem. Kontrakt se používá k modelování smlouvy v právním smyslu, ale také pro znázornění neformální dohody související s výrobkem. Kontrakt může reprezentovat Service-level agreement (SLA), která popisují dohodu o funkčnosti a kvalitě služeb, které jsou součástí produktu. Pro popis názvu kontraktu se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 71 Byznys kontrakt (zdroj: [24], vlastní úprava)

Representation (Reprezentace)

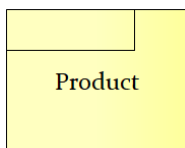
Reprezentace je charakterizována jako vnímatelná forma informací (např. zprávy nebo dokumenty), které jsou vnímatelnými nositeli informací uchovávané byznys objektem. Reprezentace je možné rozdělit dle různých skupin (např. elektronické, papírové, zvukové atd.) nebo podle formátu (HTML, PDF, RTF atd.). Pro popis reprezentace se používá podstatné jméno. (THE OPEN GROUP, c2012-2017, THE OPEN GROUP, c2012-2013)



Obrázek 72 Repräsentace (zdroj: [24], vlastní úprava)

Product (Produkt)

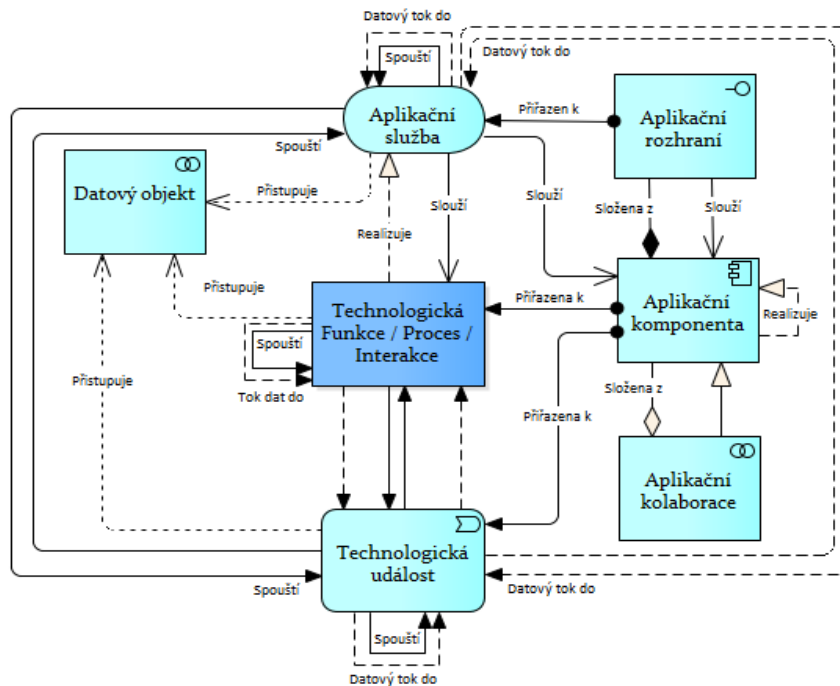
Produkt je definován jako ucelený soubor služeb, doprovázený smlouvou (kontraktem), který je nabízen jako celek interním nebo externím zákazníkům. Tato definice se vztahuje na nehmotné, finanční, služebně orientované nebo informační produkty, které jsou běžné v organizacích s vysokou intenzitou informací, ale mohou sem patřit i hmotné (fyzické) produkty. Finanční nebo informační produkt se skládá ze souboru služeb a smluv (kontraktu), které specifikují vlastnosti, práva a požadavky spojené s produktem. Pro popis názvu produktu se používá podstatné jméno (např. životní pojištění). (THE OPEN GROUP, c2012-2017)



Obrázek 73 Produkt (zdroj: [24], vlastní úprava)

3.4.2 Application layer (Aplikační vrstva)

Metamodel aplikační vrstvy vychází stejně jako metamodel byznys vrstvy z obecného modelu ArchiMate, aby ukázal přehled prvků a vazeb používaných v dané vrstvě. Stejně jako v byznys modelu nejsou v metamodelu aplikační vrstvy znázorněny všechny možné vazby mezi elementy z důvodu zachování přehlednosti modelu. Aplikační vrstva bývá propojena s byznys vrstvou pomocí aplikačních služeb, které jsou realizovány prostřednictvím softwarových aplikací.



Obrázek 74 Metamodel Aplikační vrstvy (zdroj: [24], vlastní úprava)

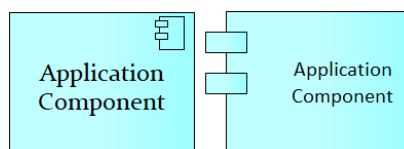
3.4.2.1 Application Structure Elements (Elementy aktivní struktury)

Hlavním prvkem aktivní struktury je aplikační komponenta (application component). Komponenta se používá k modelování jakékoli strukturní entity v aplikační vrstvě. Nejedná se pouze o (znovupoužitelné) softwarové komponenty, které jsou součástí jedné nebo více aplikací, ale aplikační komponentu lze také použít pro modelování celých aplikací, dílčích aplikací nebo informačních systémů. Aplikační komponenta v jazyce ArchiMate je velmi podobná komponentě jazyka UML, na rozdíl od toho však komponenta jazyka ArchiMate se používá výhradně pro modelování struktury aplikace, jejíž aktivity jsou znázorněny pomocí elementů chování (Behavior element). V aplikační architektuře jsou také velmi důležité vztahy mezi jednotlivými komponentami. Z toho důvodu je do jazyka ArchiMate vložena aplikační spolupráce (application collaboration), která reprezentuje soubor spolupracujících aplikačních komponent. Tento element je také inspirován kolaborací z jazyka UML.

Application Component (Aplikační komponenta)

Aplikační komponenta představuje modulární a vyměnitelnou část softwarového systému, který zapouzdřuje své aktivity a data a tuto funkcionalitu poskytuje

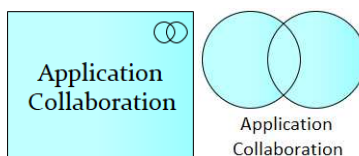
svému okolí pomocí jednoho nebo více rozhraní. Aplikační komponenta je samostatná a uzavřená jednotka, která provádí jednu nebo více funkcí aplikace a je možné jí nezávisle nasazovat, nahrazovat nebo znovu používat. Svou funkcionalitu poskytuje svému okolí prostřednictvím jednoho nebo více aplikačních rozhraní. Pro popsání aplikační komponenty je vhodné používat podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 75 Aplikační komponenta (zdroj: [24], vlastní úprava)

Application Collaboration (Aplikační součinnost)

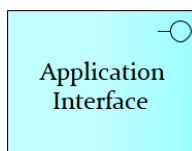
Aplikační součinnost představuje spojení dvou nebo více aplikačních komponent, které spolupracují na vykonávání společných aktivit. Aplikační kolaboraci není možné používat jako samostatný element v podniku a často při využití bývá časové omezená (existuje tak dlouho, dokud není splněn její účel). Pro popis součinnosti se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 76 Aplikační součinnost (zdroj: [24], vlastní úprava)

Application Interface (Aplikační rozhraní)

Aplikační rozhraní představuje jakýsi přístupový bod, jehož prostřednictvím je aplikační služba zpřístupněna pro použití uživatelem, aplikační komponentu nebo uzem (uzel bude představen v technologické vrstvě). Stejná aplikační služba může být poskytována prostřednictvím více aplikačních rozhraní a jedno rozhraní může být využíváno více službami. Pro popis aplikačního rozhraní je vhodné využívat podstatného jména. (THE OPEN GROUP, c2012-2017)



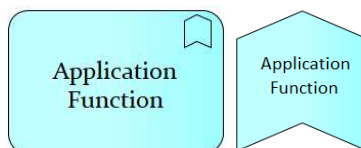
Obrázek 77 Aplikační rozhraní (zdroj: [24], vlastní úprava)

3.4.2.2 Behavior Elements (Elementy chování)

U elementů chování se podobně jako v byznys vrstvě rozlišuje mezi externími a interními aktivitami, které vykonávají aplikační komponenty. Externí aktivity aplikační komponenty jsou reprezentovány pomocí aplikačních služeb, zatímco interní aktivity představují aplikační funkce, které tyto služby realizují.

Application Function (Aplikační funkce)

Aplikační funkce reprezentuje automatizované aktivity, které mohou být vykonány aplikační komponentou. Aplikační funkce popisuje interní chování aplikační komponenty. V případě, že jsou funkce dostupné pro externí prostředí, jsou aktivity aplikační komponenty zpřístupněny pomocí jedné nebo více aplikačních služeb. Pro popis aplikační funkce se používá gerundium (slovesné podstatné jméno). (THE OPEN GROUP, c2012-2017)

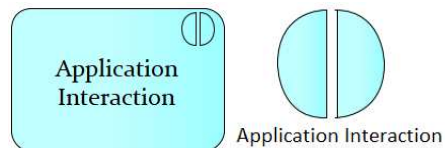


Obrázek 78 Aplikační funkce (zdroj: [24], vlastní úprava)

Application interaction (Aplikační interakce)

Aplikační interakce představuje jednotku kolektivního chování aplikací vykonávanou (spoluprací) dvou nebo více aplikačních komponent.

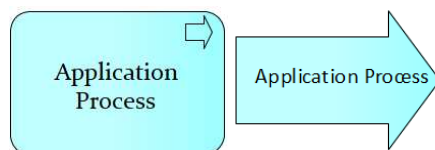
Aplikační interakce popisuje kolektivní chování (aktivity) vykonávané aplikačními komponentami, které jsou součástí aplikační součinnosti. Interakce může specifikovat externě viditelné aktivity, jež jsou nezbytné k realizaci aplikační služby. Pro popis aplikační interakce je používáno sloveso. (THE OPEN GROUP, c2012-2017)



Obrázek 79 Aplikační interakce (zdroj: [24], vlastní úprava)

Application Process (Aplikační proces)

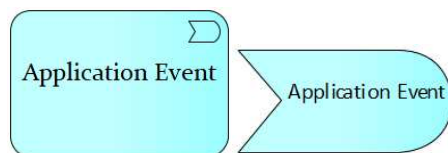
Aplikační proces je definován jako seskupení po sobě jdoucích aktivit aplikace, které mají dosáhnout určitého výsledku. Proces popisuje chování interních aktivit vykonávané aplikační komponentou, které jsou nezbytné pro realizaci souboru služeb. Pro uživatele (nezáleží, jestli se jedná o člověka nebo systém) jsou důležité služby, jaké daná aplikace nabízí. Jednotlivé aktivity, které jsou aplikací vykonávány, bývají pro uživatele nedůležité, jsou pouze černou skříňkou. Proto se nazývají interní. Procesy by měl jasně odrážet aktivitu, jakou proces vykonává (např. aktualizace údajů zákazníka nebo proces automatické aktualizace). (THE OPEN GROUP, c2012-2017)



Obrázek 80 Aplikační proces (zdroj: [24], vlastní úprava)

Application Event (Aplikační událost)

Aplikační událost je definována jako element chování aplikace, který označuje změnu stavu. Aplikační funkce a jiné elementy chování aplikace mohou být spuštěny nebo ukončeny aplikační událostí. Také elementy chování aplikační vrstvy mohou vytvářet události, které vyvolají jiné elementy chování aplikace. Na rozdíl od elementů jako jsou procesy, funkce nebo interakce, je událost okamžitá a nemá žádný časový úsek, po který je vykonávána. Jednotlivé události mohou vzniknout jako externí (např. systém běžící na cloudu u poskytovatele), nebo jako interní událostí generované jinými aplikacemi přímo v organizaci (např. se může jednat o událost, která spouští opakující se byznys proces, jenž je proveden každé první pondělí v měsíci). Pro popis elementu aplikačních událostí se používá sloveso v minulém čase. (THE OPEN GROUP, c2012-2017)

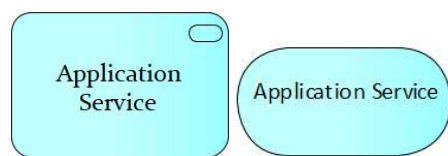


Obrázek 81 Aplikační událost (zdroj: [24], vlastní úprava)

Application Service (aplikační služba)

Aplikační služba poskytuje svému prostředí funkcionalitu aplikační komponenty, ke které je přístupováno pomocí jednoho nebo více aplikačních rozhraní. Aplikační služba poskytuje jednotku chování, která má z pohledu prostředí určitý význam a je sama o sobě užitečná pro uživatele (může jím být i aplikační komponenta nebo uzel).

Pro popis aplikační služby se používá podstatné jméno slovesné (např. zpracování žádostí). V názvu je možné používat slovo „služba“. (THE OPEN GROUP, c2012-2017)



Obrázek 82 Aplikační služba (zdroj: [24], vlastní úprava)

3.4.2.3 Passive Structure Elements (Elementy pasivní struktury)

V oblasti pasivní struktury se nachází pouze jeden element, kterým je datový objekt. Tento element se používá stejným způsobem jako datové objekty (nebo typy objektů) v diagramu tříd jazyka UML. Jazyk ArchiMate nedefinuje oddělenou vrstvu pro určování informací, proto jsou informace reprezentovány pomocí elementů ostatních vrstev, jako jsou byznys objekty a datové objekty. (THE OPEN GROUP, c2012-2017)

Data Object (Datový objekt)

Datový objekt je definován jako strukturovaná data vhodná pro automatické zpracování aplikací. Datový objekt by měl obsahovat samostatnou informaci s jasným významem pro firmu, a to nejenom na aplikační úrovni. Typický příkladem datového objektu je záznam o partnerech nebo klientská databáze. Pro

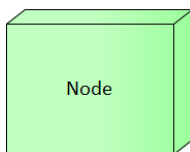
3.4.3.1 Active Structure Elements (Elementy aktivní struktury)

Hlavním elementem aktivní struktury technologické vrstvy je uzel (node), který reprezentuje výpočetní zdroj sloužící pro uložení artefaktů nebo práce s nimi. Uzel se používá výhradně k modelování struktury systému a jeho aktivity jsou modelovány výlučně pomocí vazeb na odpovídající elementy chování.

Node (Uzel)

Uzel je definovaný jako výpočetní nebo fyzický zdroj, který umožňuje uložení, manipulaci nebo spolupráci s jinými výpočetními nebo fyzickými prostředky.

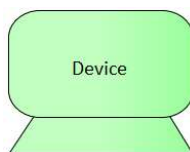
Uzel je element aktivní struktury, který provádí technologické aktivity, usnadňující nebo zpracovávající technologické objekty jako jsou artefakty. Uzel se může skládat z dílčích uzlů a také je často kombinací hardwaru zařízení a systémového softwaru, jež dohromady poskytují své služby pro podporu a chod aplikací. Pro popis uzlů se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 85 Uzel (zdroj: [24], vlastní úprava)

Device (Zařízení)

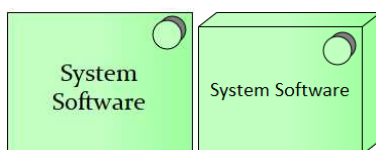
Zařízení je definováno jako fyzický zdroj IT, na kterém mohou být systémový software a artefakty uloženy nebo nasazeny pro zpracování. Zařízení je specializací uzlu. Obvykle se tento element používá k modelování hardwarových systémů, jako jsou mainframe (sálové počítače), počítače nebo směrovače. Zařízení je nejčastěji součástí uzlu společně se systémovým softwarem. Pro popis elementů zařízení se používá podstatné jméno odkazující na typ hardwaru (např. DELL sálový počítač). (THE OPEN GROUP, c2012-2017)



Obrázek 86 Zařízení (zdroj: [24], vlastní úprava)

System software (Systémový software)

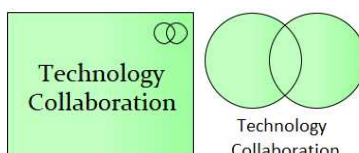
Systémový software představuje softwarové prostředí, které napomáhá k ukládání, provádění nebo používání softwaru nebo dat v něm nasazených. Systémový software je specializací uzlu, který se používá k modelování softwarového prostředí, v němž běží artefakty. Mezi systémový software může např. patřit operační systém, databázový systém, firmware nebo middleware. Pro popis elementu systémového softwaru se používá podstatné jméno, jež by mělo odrážet systémového prostředí (např. DBMS server). (THE OPEN GROUP, c2012-2017)



Obrázek 87 Systémový software (zdroj: [24], vlastní úprava)

Technology Collaboration (Technologická součinnost)

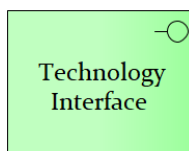
Technologická součinnost představuje spojení dvou nebo více uzlů, které spolupracují na vykonávání společných aktivit. Kolaboraci není možné používat jako samostatný element v podniku a často při využívání vytváří logické nebo časově omezené spojení dvou uzlů (existuje tak dlouho, dokud není splněn jeho účel). Pro popis součinnosti se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 88 Technologická součinnost (zdroj: [24], vlastní úprava)

Technology Interface (Technologické rozhraní)

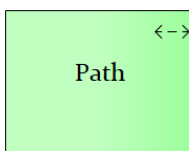
Technologické rozhraní představuje přístupový bod, jehož prostřednictvím se lze dostat k službám nabízeným uzlem. Díky tomu je možné, aby jeden uzel přistupoval ke službám jiného uzlu nebo nabízel své služby okolnímu prostředí (např. aplikační komponentě). Pro název technologického rozhraní se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 89 Technologické rozhraní (zdroj: [24], vlastní úprava)

Path (Cesta)

Cesta reprezentuje spojení mezi dvěma nebo více uzly, prostřednictvím kterého mohou uzly vyměňovat data nebo materiál (materiál bude popsán v části Fyzické rozšíření). Cesta je realizována pomocí jedné nebo víc komunikačních sítí (nebo distribučních sítí, jsou-li modelovány fyzické prvky z části Fyzického rozšíření), které reprezentují fyzickou komunikaci (distribuci). (THE OPEN GROUP, c2012-2017)



Obrázek 90 Cesta (zdroj: [24], vlastní úprava)

Communication Network (Komunikační síť)

Komunikační síť reprezentuje fyzickou komunikační infrastrukturu (představuje propojení systémů nebo zařízení za účelem přenosu dat mezi systémy), která se může skládat ze dvou nebo více zařízení. Nejjednodušší komunikační síť je jediné spojení mezi dvěma zařízeními, avšak komunikační síť může obsahovat více propojení a přidružených síťových zařízení. Síť může mít vlastnosti jako šířka pásma (bandwidth) nebo odezva (latency). Příkladem je směrovač, switch nebo firewall. (THE OPEN GROUP, c2012-2017)



Obrázek 91 Komunikační síť (zdroj: [24], vlastní úprava)

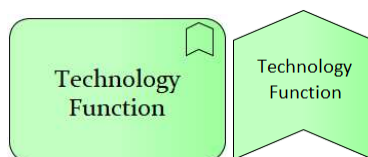
3.4.3.2 Behavior Elements (Elementy chování)

Elementy chování technologické vrstvy jsou značně podobné elementům z byznys a aplikační vrstvy. Také v této vrstvě dochází k rozlišování mezi externími aktivitami uzlů (technologické služby) a interními aktivitami uzlů (technologické

funkce). Funkce představují seskupení technologických aktivit, které uzel vykonává. Technologická služba je externě viditelnou funkcionalitou poskytovanou jednomu nebo více uzlům.

Technology Function (Technologická funkce)

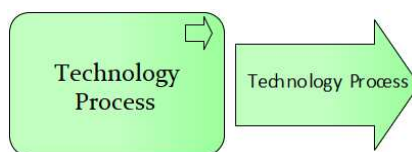
Technologická funkce představuje seskupení technologických aktivit, které může uzel vykonávat. Tato funkce popisuje interní aktivity uzlu, které jsou pro uživatele uzlu neviditelné. Pokud jsou aktivity uzlu veřejně dostupné, je tak učiněno prostřednictvím jedné nebo více technologických služeb. Pro popis technologické funkce se používá gerundium (slovesné podstatné jméno). (THE OPEN GROUP, c2012-2017)



Obrázek 92 Technologická funkce (zdroj: [24], vlastní úprava)

Technology Process (Technologický proces)

Technologický proces je definován jako posloupnost technologických aktivit, které mají dosáhnout určitého výsledku. Proces může využívat technologické objekty jako svůj výstup nebo používat jiné objekty jako vstup a dále je transformovat a vytvářet tak nové technologické objekty. Název procesu by měl jasně identifikovat řadu technologických aktivit (např. obnovení databáze, zavedení systémové posloupnosti). (THE OPEN GROUP, c2012-2017)

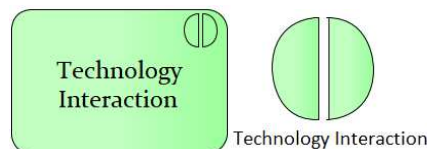


Obrázek 93 Technologický proces (zdroj: [24], vlastní úprava)

Technology Interaction (Technologická interakce)

Technologická interakce představuje jednotku kolektivního chování vykonávanou spoluprací dvou nebo více uzlů. Technologická interakce popisuje kolektivní chování (aktivity) vykonávané spolupracujícími uzly. Interakce může také specifikovat externě viditelné aktivity, které jsou nezbytné k realizaci

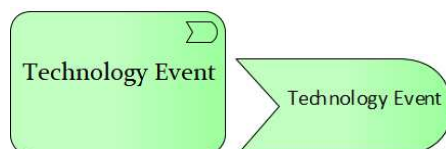
technologických služeb. Pro popis technologické interakce se používá sloveso (např. aktualizace záznamu o pracovnících, vytvoření profilu klienta). (THE OPEN GROUP, c2012-2017)



Obrázek 94 Technologická interakce (zdroj: [24], vlastní úprava)

Technology Event (Technologická událost)

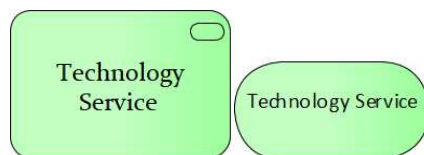
Technologická událost je definována jako element chování technologické vrstvy, která označuje změnu stavu. Technologické funkce a jiné technologické elementy chování mohou být spuštěny nebo přerušeny technologickou událostí. Také technologická funkce může vyvolat událost, která spustí jiné elementy chování technologické vrstvy. Na rozdíl od elementů jako jsou procesy, funkce nebo interakce, je událost okamžitá a nemá časový úsek, ve kterém je vykonávána. Události mohou pocházet z okolního prostředí organizace (externí události) nebo mohou vznikat interní události generováním pomocí jiných zařízení v organizaci (např. událost spouštěcí funkci zálohování dat, která se provede každý den). Pro popis elementu technologické události se používá sloveso v minulém čase (např. zpráva přijata). (THE OPEN GROUP, c2012-2017)



Obrázek 95 Technologická událost (zdroj: [24], vlastní úprava)

Technology Service (Technologická služba)

Technologická služba poskytuje svému prostředí funkcionalitu uzlu, ke kterému je přístupováno pomocí jednoho nebo více technologických rozhraní. Technologická služba může používat a vytvářet artefakty. Mezi typické služby patří například služba pro zasílání zpráv, poskytování úložného prostoru nebo adresářové služby. Pro popis elementu technologické služby se používá gerundium (slovesné podstatné jméno, např. poslání zprávy). V názvu elementu je také povoleno používat přímo slovo „služba“. (THE OPEN GROUP, c2012-2017)



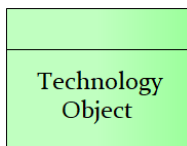
Obrázek 96 Technologická služba (zdroj: [24], vlastní úprava)

3.4.3.3 Passive Structure Elements (Elementy pasivní struktury)

Pasivní struktura technologické vrstvy obsahuje dva elementy. Prvním je technologický objekt, který představuje fyzický prvek (např. materiál). Druhým elementem pasivní struktury je artefakt, který reprezentuje fyzické informace používané nebo produkované při procesu vývoje softwaru nebo při implementaci a provozu systému. Tento element je převzat z jazyka UML. (THE OPEN GROUP, c2012-2017)

Technology Object (Technologický objekt)

Technologický objekt je definován jako fyzický objekt, který se využívá nebo vytváří při technologických aktivitách. Tento objekt zastupuje obecný typ věcí (tedy není konkrétní), se kterými je manipulováno v rámci elementů technologické vrstvy. Pro popis objektů se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)

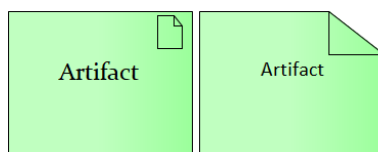


Obrázek 97 Technologický objekt (zdroj: [24], vlastní úprava)

Artifact (Artefakt)

Artefakt je definován jako data používaná nebo vytvářená při procesu vývoje softwaru nebo při implementaci a provozu systému. Artefakt reprezentuje konkrétní element v reálném světě a obvykle se používá k modelování softwarových produktů, mezi které patří zdrojové kódy, spustitelné soubory, skripty, databázové tabulky, zprávy dokumenty, specifikace a soubory modelů. Artefakt může nabývat dvou stavů, a to jako komponenta určená pro zpracování nebo jako datový soubor.

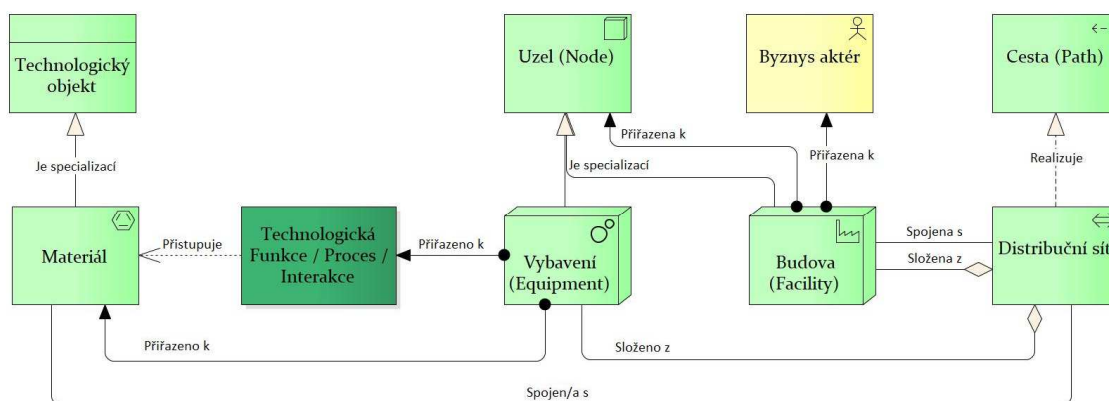
Pro popis artefaktu se používá název souboru (např. objednávka.zip, databáze_zaměstnanců.jar). (THE OPEN GROUP, c2012-2017)



Obrázek 98 Artefakt (zdroj: [24], vlastní úprava)

3.4.4 Physical Elements (Fyzické elementy)

Fyzické elementy jsou jakýmsi rozšířením technologické vrstvy. Doplnují chybějící elementy v předchozí verzi jazyka ArchiMate, např. prostředí (facility). Poskytují prostory pro vykonávání aktivit nebo uskladnění materiálu, ale také vybavení, kterým jsou myšleny stroje, rovněž distribuční sítě, jež mohou znázorňovat přepravu materiálu. Na obrázku níže je uveden metamodel fyzických elementů, který vychází především z technologické vrstvy, ale zde se vyskytnou i návaznosti na ostatní vrstvy. Metamodel neobsahuje z důvodu zachování přehlednosti všechny povolené vazby mezi použitými elementy. (THE OPEN GROUP, c2012-2017)



Obrázek 99 Metamodel Fyzických prvků (zdroj: [24], vlastní úprava)

3.4.4.1 Active Structure Elements (Elementy aktivní struktury)

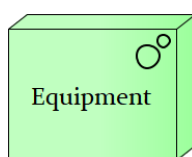
Element vybavení (equipment) je hlavním prvkem aktivní struktury fyzických elementů. Vybavení se používá k znázornění fyzických strojů, nástrojů, přístrojů nebo nářadí. Tento element se používá k modelování struktury systému a jeho aktivity jsou modelovány pomocí vazeb na odpovídající elementy chování.

Vzájemné propojení mezi fyzickými elementy slouží především k vytváření logistické infrastruktury. Cesta z technologické vrstvy modeluje vztahy mezi dvěma nebo více uzly, prostřednictvím kterých mohou přenášet informace nebo materiál. Pro znázornění fyzické cesty se používá element distribuční sítě, který znázorňuje fyzické spojení mezi dvěma nebo více vybavením (equipment) nebo jinými fyzickými sítěmi (cestami), jako např. železniční nebo silniční sítě, dodávky vody, elektrické sítě atd.

Equipment (Vybavení)

Vybavení je definováno jako jeden nebo více fyzických zařízení (stroje, nářadí atd.), které mohou vytvářet, používat, přesouvat nebo jinak pracovat s materiálem.

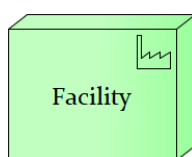
Vybavení se může skládat z aktivních elementů provádějících fyzické procesy, ve kterých se používá nebo transformuje materiál. Jako příklad může být uvedena magnetická rezonance používaná v nemocnicích, která je sama o sobě vybavením využívající specifický software. Pro popis vybavení se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 100 Vybavení (zdroj: [24], vlastní úprava)

Facility (Prostředí)

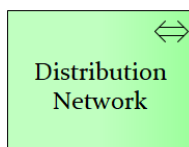
Prostředí je definováno jako fyzický objekt, který má schopnost uschování nebo umístění používaného vybavení (equipment). Obvykle se používá k modelování továren, budov nebo venkovních konstrukcí, které hrají důležitou roli ve výrobním nebo distribučním procesu. Pro popis prostředí se používá podstatného jména, které by mělo jasně definovat budovu nebo prostředí, které reprezentuje (např. Rotterdamská přístavní rafinerie). (THE OPEN GROUP, c2012-2017)



Obrázek 101 Prostředí (zdroj: [24], vlastní úprava)

Distribution Network (Distribuční síť)

Distribuční síť je definovaná jako fyzická cesta sloužící k přepravě materiálu nebo energie. Distribuční síť představuje fyzickou distribuci nebo dopravní infrastrukturu, fyzicky realizuje logické cesty mezi uzly. Jako příklad může být uvedena železniční stanice s vlaky, které jsou součástí železniční sítě. (THE OPEN GROUP, c2012-2017)



Obrázek 102 Distribuční síť (zdroj: [24], vlastní úprava)

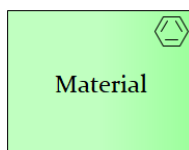
3.4.4.2 Behavior Elements (Elementy chování)

V této části nejsou definovány žádné fyzické elementy chování. Jelikož jsou fyzické elementy velmi úzce spojeny s technologickou vrstvou, jsou i elementy chování používány z této vrstvy (technologická funkce, proces, interakce, služba nebo událost). Ty používají k modelování aktivit uzlů včetně fyzického zařízení. Vzhledem k tomu, že většina vybavení (equipment) je velmi často řízena počítačem nebo jiným způsobem, který má blízko k IT (internet věcí), jejich chování může být popsáno s využitím elementů chování technologické vrstvy. (THE OPEN GROUP, c2012-2017)

3.4.4.3 Passive Structure Elements (Elementy pasivní struktury)

Material (Materiál)

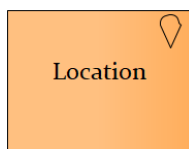
Materiál představuje hmatatelný nebo fyzický element s atributy jako je velikost a váha. Obvykle se používá k modelování surovin nebo fyzických produktů (např. palivo, zboží na skladě). Element reprezentující materiál se může skládat z více kusů, kde každou část bude představovat jiný grafický element. Pro popis materiálu se používá podstatné jméno. (THE OPEN GROUP, c2012-2017)



Obrázek 103 Materiál (zdroj: [24], vlastní úprava)

Location (Lokace)

Lokace je definovaná jako místo (fyzické nebo virtuální), ve kterém se dané elementy, jež zaštiťuje, nacházejí. Lokace se používá k vyjádření umístění strukturálních elementů, jako jsou byznys aktéři, aplikační komponenty nebo zařízení. (THE OPEN GROUP, c2012-2017)

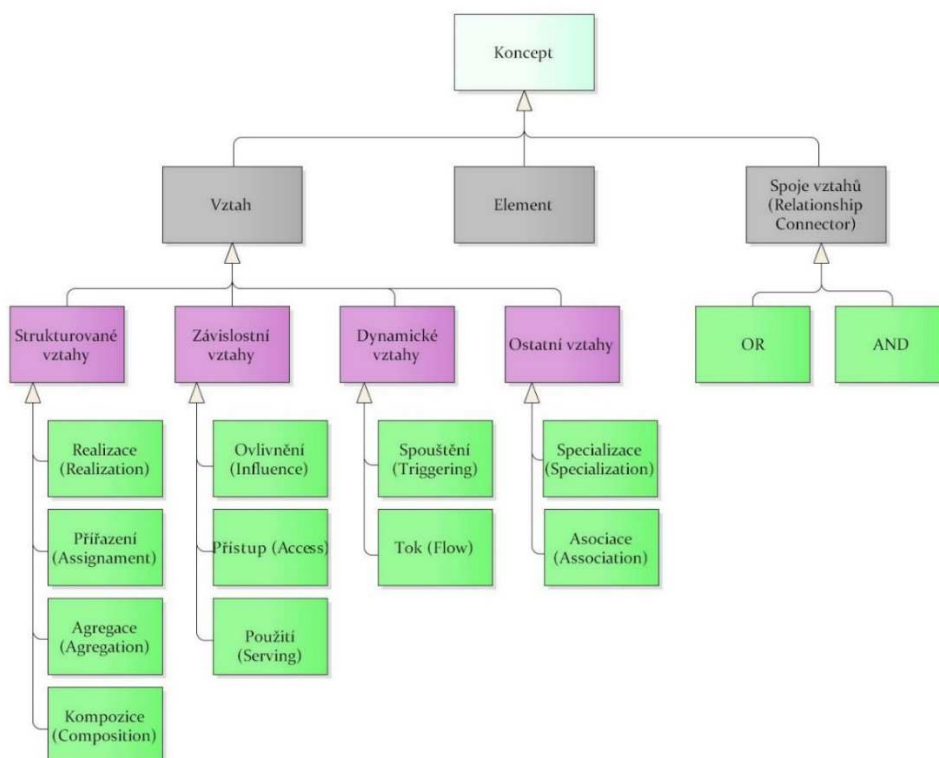


Obrázek 104 Lokace (zdroj: [24], vlastní úprava)

3.4.5 Vazby (vztahy) mezi elementy

ArchiMate rozděluje vztahy mezi elementy do těchto 4 kategorií:

- Strukturální vztahy – jedná se o vztahy, které popisují spojitosti mezi elementy stejného nebo odlišného typu.
- Závislostní vztahy – popisuje, jakým způsobem jsou podpůrné elementy závislé na podporovaných elementech.
- Dynamické vztahy – používají se k popisu závislostí mezi elementy chování.
- Ostatní vztahy – vazby, které není možné zařadit do některé z výše uvedených kategorií.



Obrázek 105 Přehled vztahů (zdroj: [24], vlastní úprava)

Každá vazba se může nacházet pouze mezi dvěma elementy. Tedy to znamená, že má vždy jeden bod, od kterého vychází a pouze jeden ke kterému přichází. Pro vazby dále platí tato omezení:

- Mezi dvěma vztahy nejsou povoleny žádné jiné vztahy (dvě vazby nemohou být mezi sebou propojeny jinou vazbou).
- Veškeré vztahy propojené se spojem vztahů (Relationship connector – v jazyce BPMN známe jako gateway) musí být stejného typu.
- Řetězec vztahů stejného typu spojující dva elementy prostřednictvím Relationship Connectors, je platný pouze tehdy, je-li přímý vztah stejného typu vztahů mezi těmito dvěma elementy platný. (To znamená, že Relationship Connector můžeme použít vždy, pokud daná vazba je mezi dvěma elementy povolená).
- Mezi elementy stejného typu je možné používat pouze vazby agregace, kompozice a asociace.

Z důvodu zachování přehlednosti metamodelu v tomto dokumentu nejsou zobrazeny všechny možné vztahy mezi elementy. V další kapitole budou popsány pravidla pro odvození nepřímých vztahů, které se do modelu nevkládají z důvodu jeho přehlednosti a čitelnosti.

3.4.5.1 Strukturální vztahy

Strukturální vztahy představují v architektuře staticky soudržné vazby, tedy takové vazby, které se v čase nemění. Vazby v modelu spojují vždy dva elementy mezi sebou. A to od jednoho elementu k druhému. V případě složitějších modelů může být element dalším vztahem. To znamená, že vazba jde od jednoho elementu k druhému a od druhého pokračuje další vztah (vazba). Jako alternativa ke grafickému zápisu vazeb navrženém v této kapitole, mohou být strukturované vztahy vyjádřené prostřednictvím vnořených elementů. V některých případech je však důležité se předem rozhodnout, jaký typ grafického znázornění bude použit. Špatná volba by mohla vést k nejednoznačnosti modelu, a to především je-li v něm použito velké množství elementů nebo mezi elementy je povoleno více strukturálních vazeb. Nepsané pravidlo říká: „Pokud máme k jednomu elementu připojeno více jak pět elementů vazbou agregace nebo kompozice, měly by se tyto elementy použít jako vnořené.“ (THE OPEN GROUP, c2012-2017)

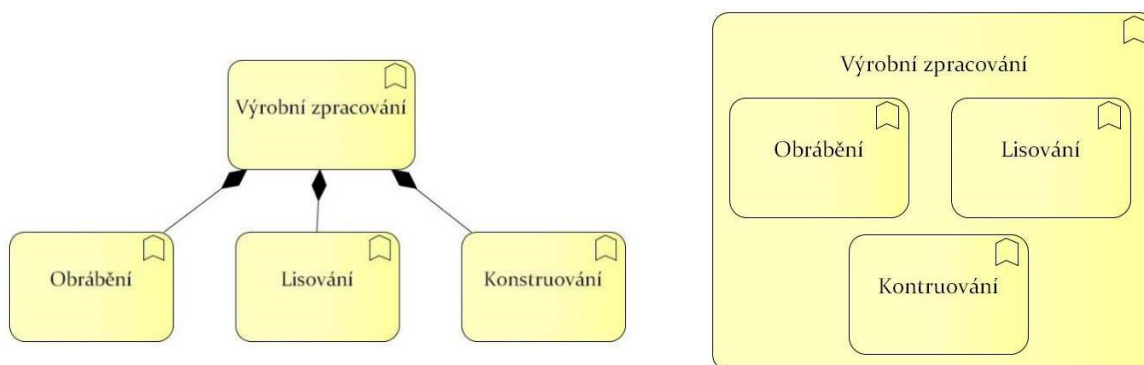
Composition (Kompozice)

Kompozice označuje situaci, kdy se element skládá z jednoho nebo více elementů. Kompozice pochází z jazyka UML, kde se používá v class diagramu (diagramu tříd). Tento vztah se nejčastěji aplikuje pouze mezi dvěma elementy stejného typu. Obvyklá interpretace kompozice je, že dílčí element je součástí hlavního elementu, a tudíž pokud dojde k zániku některého z elementů, zbylé elementy zanikají. Na rozdíl od agregace může být element součástí pouze jedné kompozice. (THE OPEN GROUP, c2012-2017)



Na obrázku je možné vidět grafické notace kompoziční vazby pomocí černého plného kosočtverce a čáry.

Obrázek 106 Kompozice (zdroj: [24], vlastní úprava)



Obrázek 107 Příklad vazby kompozice (zdroj: vlastní tvorba)

Na obrázku výše jsou znázorněny dva způsoby, jak vyjádřit agregační vztah mezi elementy. V levé části obrázku můžeme vidět klasické propojení elementů pomocí vazby. V pravé části obrázku je znázorněn kompoziční vztah pomocí vnořených elementů. Obě tyto varianty jsou si rovnocenné, a tedy vypovídají o tom, že výroba nemůže pokračovat, pokud by některá ze tří funkcí nebyla provozuschopná.

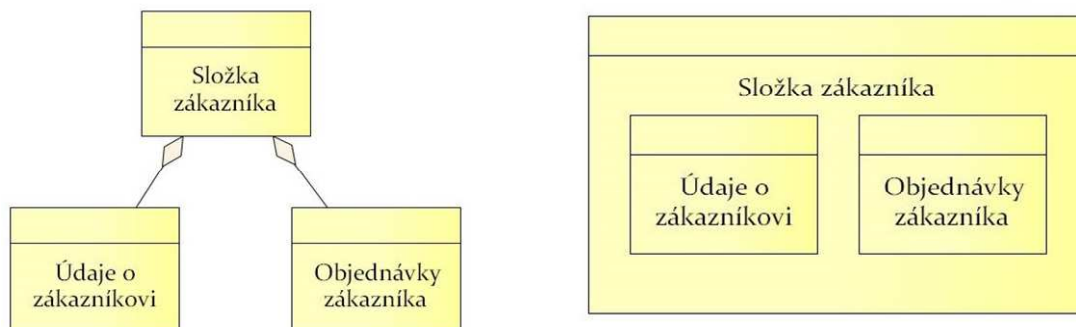
Aggregation (Agregace)

Agregace je vazba, která reprezentuje situaci, kdy element seskupuje určitý počet dalších elementů. Na rozdíl od kompoziční vazby může být element součástí více než jedné agregace. Agregace stejně jako kompozice se nejčastěji používá mezi vazbami stejného typu. Na obrázku je zachycena grafická notace agregační vazby

znázorněná pomocí prázdného kosočtverce s čarou. (THE OPEN GROUP, c2012-2017)



Obrázek 109 Agregace (zdroj: [24], vlastní úprava)



Obrázek 108 Příklad vazby agregace (zdroj: vlastní tvorba)

Na obrázku výše jsou stejně jako u příkladu kompozice zachyceny dva způsoby grafického znázornění, a to spojení elementů pomocí vazby nebo vnořím elementů. Agregace se liší od kompozice zejména tím, že pokud zaniknou některé elementy, ostatní mohou existovat (jestliže jsou smazány zákaznickovy objednávky, složka s údaji může dále existovat).

Assignment (Přiřazení)

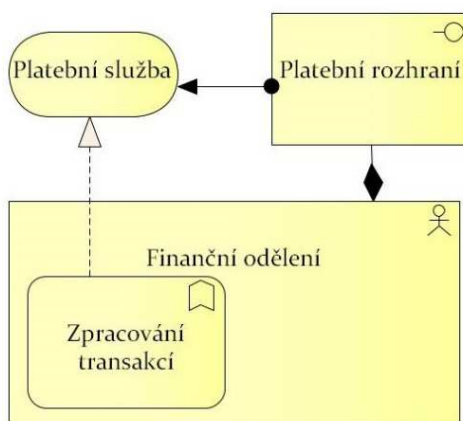
Přiřazení vyjadřuje přidělení odpovědnosti, chování nebo provedení nějaké činnosti. Přiřazení spojuje elementy aktivní struktury s elementy chování, které jsou jimi vykonávány. Příkladem může být byznys aktér spojený s byznys rolí, kterou ve firmě zastává. Podobně jako u ostatních strukturálních vztahů, může být vztah přiřazení také vyjádřen pomocí vnoření elementů. V případě využití vazby je směr přiřazení jasný, udává ho šipka. V případě vnoření elementů má vždy vrchní element přiřazen element vnořený. Příkladem může být byznys funkce vložená do elementu byznys aktéra, který tuto funkci vykonává. (THE OPEN GROUP, c2012-2017)

Na obrázku je zachycena grafická notace Assignment (Přiřazení):



Obrázek 110 Přiřazení (zdroj: [24], vlastní úprava)

Obrázek 111 obsahuje příklad popisující dva způsoby, jak je možné vyjádřit vazbu přiřazení. V horní části příkladu je ukázáno přiřazení pomocí vazby, kde je byznys rozhraní (platební rozhraní) k byznys službě (platební služba). V spodní části obrázku je finanční oddělení, které má přiřazeno funkci zpracování transakcí, pomocí vnoření elementů.



Obrázek 111 Příklad vazby přiřazení (zdroj: vlastní tvorba)

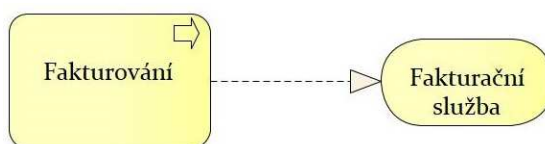
Realization (Realize)

Realizační vztah naznačuje, že entita hraje rozhodující roli při vytváření, dosažení, udržování nebo provádění více specifické entity. Realizace naznačuje, jak jsou logické entity (na které se můžeme zeptat prostřednictvím otázky "co realizují?") realizovány prostřednictvím více specifických entit (na které se můžeme ptát skrze dotaz "jak to realizují?"). Realizace se využívá pro znázornění vztahů, které jsou důležité pro chod systému: např., že byznys proces realizuje byznys službu a datový objekt realizuje byznys objekt, nebo že artefakt realizuje aplikační komponentu. (THE OPEN GROUP, c2012-2017)



Obrázek 112 Realizace (zdroj: [24], vlastní úprava)

Obrázek níže znázorňuje použití vazby realizace, kdy je aplikován proces fakturování, od kterého jde šipka realizace k fakturační služba. Tento vztah se čte tak, že fakturační služba je realizována pomocí procesu fakturování.



Obrázek 113 Příklad vazby realizace (zdroj: vlastní tvorba)

Pro znázornění slabšího vztahu mezi elementy než je realizace, je možné tuto vazbu nahradit vztahem ovlivnění (Influence).

3.4.5.2 Závislostní vazby (vztahy)

Závislostní vazby popisují, jak elementy ovlivňují jiné elementy nebo jsou jimi ovlivňovány. Závislostní vztahy se dělí do tří skupin:

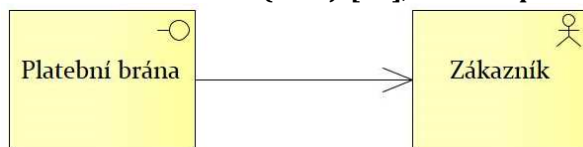
- Obslužný vztah (Serving relationship), ve starších verzích ArchiMate aplikován také jako „vazba použití“ znázorňující řídicí funkcionalitu (závislosti) mezi elementy. Tato vazba je označována pevnou čarou.
- Přístupový vztah (Access relationship) představuje především závislost mezi elementy a datovými soubory. Pro znázornění tohoto vztahu je využíváno přerušované čáry.
- Vztah ovlivnění (Influence relationship) je jeden z nejslabších závislostních vazeb, který je používán pro znázornění, jak jsou motivační elementy ovlivňovány jinými elementy.

Serving (Sloužící)

Tento vztah popisuje, jak jsou služby nebo rozhraní nabízené prvky chování nebo aktivní struktury používány okolním prostředím. Ve srovnání se starší verzí jazyka ArchiMate byl název této vazby změněn z „používá“ na „slouží“, aby lépe odrážel význam tohoto vztahu. Vezmeme-li starší význam vazby a spojili bychom s ní uživatele a rozhraní, četlo by se, že rozhraní je používána uživatelem, kdežto nový význam vazby říká, že rozhraní slouží uživateli. Oba ze zmíněných příkladů jsou povolené používat, ale doporučováno je používat novější význam. (THE OPEN GROUP, c2012-2017)



Obrázek 114 Sloužící (zdroj: [24], vlastní úprava)

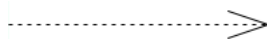


Obrázek 115 Příklad sloužící vazby (zdroj: vlastní tvorba)

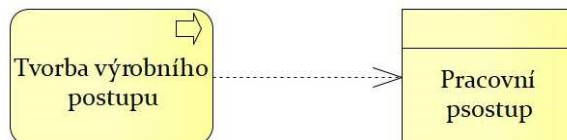
Access (Přístup)

Tento vztah modeluje schopnost elementů chování nebo aktivní struktury přistupovat, vytvářet nebo upravovat elementy pasivní struktury. Přístupová

vazba znázorňuje, kdy proces, funkce, interakce služba nebo událost manipuluje s pasivními elementy struktury (např. vytváření datového objekt, čtení data z objektu, úprava dat objektu, odstranění objektu). Pokud je u vazby šipka, znázorňuje směr toku dat. Je-li potřeba znázornit obousměrný tok dat, šipka se nepoužívá. (THE OPEN GROUP, c2012-2017)



Obrázek 116 Přístup (zdroj: [24], vlastní úprava)



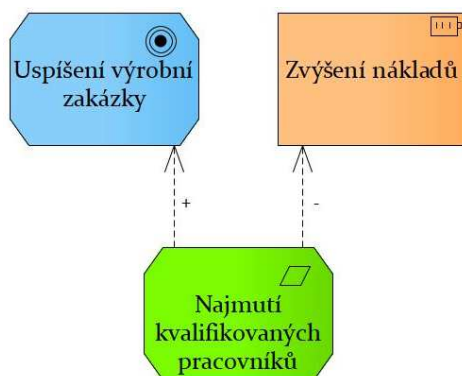
Obrázek 117 Příklad vazby přístup (zdroj: vlastní tvorba)

Influence (Ovlivnění)

Vazba influence popisuje, jak jeden motivační element může ovlivnit dosažení nebo realizaci jiného motivačního elementu. Vezměme si kupříkladu cíl, kdy firma chce dosáhnout vyšší spokojenosti zákazníků na trhu. Tento cíl může být pozitivně ovlivněn tím, že firma bude snižovat ceny svých výrobků (služeb) nebo bude poskytovat odborné konzultace. Naopak, vezme-li si firma za cíl propustit větší množství zaměstnanců, může to vést k její negativní reputaci a negativně tak ovlivnit i cíl zvýšit spokojenost zákazníka. Vazba influence tedy znázorňuje skutečnost, kdy určitý element pozitivně nebo negativně přispívá k dosažení nebo implementaci některého z motivačních elementů. Tato situace je znázorněna pomocí znaménka + (pozitivní ovlivnění), nebo symbolu – (negativní vliv). Ve výchozím nastavení vazby influence není definován symbol pozitivního (+) nebo negativního (-) ovlivnění a není povinností ho používat. (THE OPEN GROUP, c2012-2017)



Obrázek 118 Ovlivnění (zdroj: [24], vlastní úprava)



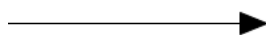
Obrázek 119 Příklad vazby ovlivnění (zdroj: vlastní tvorba)

3.4.5.3 Dynamické vztahy

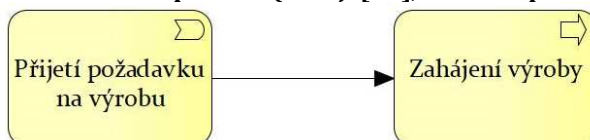
Dynamické vztahy popisují dočasné závislosti mezi elementy v architektuře. Tyto vztahy existují pouze dva: triggering (spuštění) a flow (tok).

Triggering (Spuštění)

Spuštění popisuje časové nebo kauzální závislosti mezi elementy. Spouštěcí vazba se používá k znázornění časového nebo kauzálního spuštění (vlivem nějaké příčiny) elementů chování v rámci procesu. Interpretace vazby spuštění říká, že zdrojový element by měl být dokončen dříve, než může být spuštěn cílový element. Jazyk ArchiMate nerozlišuje mezi aktivním a pasivním (kauzálním) spuštěním, a tedy nemusí nutně znamenat, že zdrojový element musí být z aktivní struktury. (THE OPEN GROUP, c2012-2017)



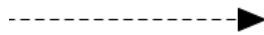
Obrázek 120 Spuštění (zdroj: [24], vlastní úprava)



Obrázek 121 Příklad vazby spuštění (zdroj: vlastní tvorba)

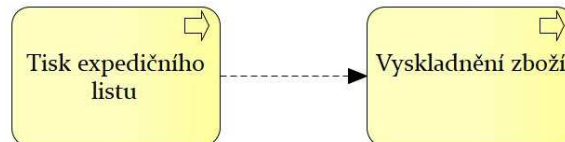
Flow (Tok)

Tok popisuje přenos z jednoho elementu na druhý. Používá se k modelování toku (např. informací, zboží nebo peněz) mezi elementy chování. Tok neslouží k vyjádření příčinného vztahu. (THE OPEN GROUP, c2012-2017)



Obrázek 122 Tok (zdroj: [24], vlastní úprava)

Obrázek znázorňuje klasický tok, kdy nejdříve musí být vytištěn expediční list, který je odeslán skladníkovi, aby zboží vyskladnil. (THE OPEN GROUP, c2012-2017)



Obrázek 123 Příklad vazby tok (zdroj: [24], vlastní úprava)

3.4.5.4 Ostatní vztahy

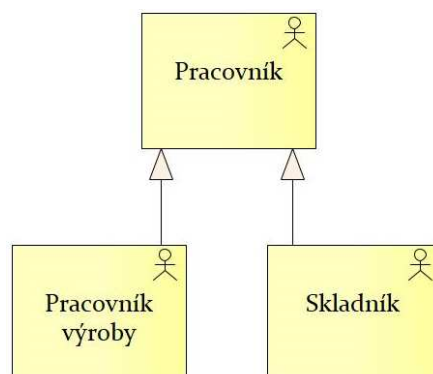
Specialization (Specializace)

Specializace vyjadřuje, že je daný element specializací jiného elementu. Vztah specializace byl inspirován vztahem generalizace z jazyka UML, ale v jazyce ArchiMate je možné jej použít u většího množství elementů. Mezi dvěma instancemi stejného prvku je vztah specializace vždy povolen. (THE OPEN GROUP, c2012-2017)



Obrázek 124 Specializace (zdroj: [24], vlastní úprava)

Z obrázku 125 vyplývá, že pracovník výroby a skladník jsou specializací pracovníka.



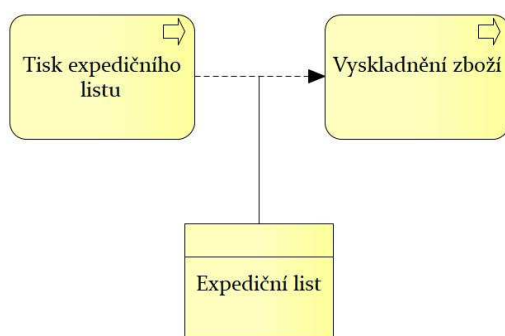
Obrázek 125 Příklad vazby specializace (zdroj: vlastní tvorba)

Association (Asociace)

Asociace se nejčastěji používá pro spojování elementů, které nemají jiný specifikovaný vztah. Asociace je povolena mezi dvěma elementy nebo mezi vazbou a elementem. Asociace se nejčastěji používá u datových objektů nebo při modelování vyšších úrovní modelu, kde není z počátku možné přesně definovat specifický typ vazby. (THE OPEN GROUP, c2012-2017)

Obrázek 126 Asociace (zdroj: [24], vlastní úprava)

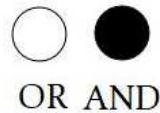
Na obrázku je znázorněn model toku (flow), který je rozšířen pomocí asociace o expediční list. Ten je výsledkem procesu tisk expedičního listu a dále vstupuje do procesu vyskladnění zboží, jako dokument o zboží, které má být vyskladněno. (THE OPEN GROUP, c2012-2017)



Obrázek 127 Příklad vazby asociace (zdroj: vlastní tvorba)

Junction (Spojka)

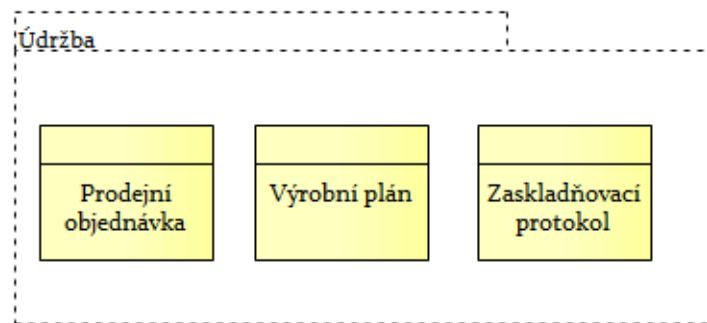
Spojka se používá v řadě situací pro spojení vztahů stejného typu. Tento vztah se aplikuje v místech, kde dochází ke spojování nebo větvení vztahů. Spojku lze používat u všech dynamických a závislostních vztahů, ale také u přiřazení, realizace a asociace. Spojka připomíná gateway z jazyka BPMN, kde plný symbol (And Junction) stejně jako (AND Gateway) spojí nebo rozdělí vždy obě cesty. Kdežto (OR Junction) je stejný jako (OR Gateway) – zde je tok spojen nebo rozdělen podle podmínky, na kterou by měla být odpověď ANO nebo NE. A podle té podmínky, která je splněna, pokračuje dále tok. (THE OPEN GROUP, c2012-2017)



Obrázek 128 Spojka (zdroj: [24], vlastní úprava)

Grouping (Seskupení)

Seskupení agreguje nebo skládá objekty, které mají společné nějaké charakteristiky. Tento element se používá pro vytvoření libovolné skupiny objektů, které mohou být stejného nebo rozdílného typu. Na rozdíl od agregačních a kompozičních vztahů neexistuje mezi elementy v seskupení nadřazený objekt, jenž by se z podřízených objektů skládal nebo je agregoval. (THE OPEN GROUP, c2012-2017)



Obrázek 129 Seskupení zdroj: vlastní tvorba)

3.5 Enterprise Architect

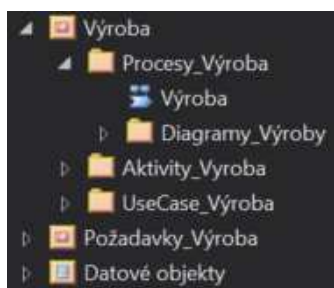
Enterprise Architect je rozsáhlý grafický nástroj, který napomáhá při návrhu systému, modelování byznys procesů nebo modelování struktury podniku. Enterprise Architect je založen především na specifikaci UML, ale podporuje i jiné jazyky jako je BPMN nebo ArchiMate. Díky tomu je možné pokrýt všechny aspekty vývojového cyklu, který ho umožní sledovat od počáteční fáze návrhu až po nasazení, údržbu, testování a kontrolu změn. I přesto, že je nástroj ideálním pomocníkem při návrhu systému, umožňuje graficky znázornit i jiné aspekty podniku, jakými jsou nesystémové úlohy, kompletní zmapování firmy, včetně služeb, firemní struktury, nebo umožňuje vytvořit pojmové mapy. Pro modelování existují i jiné nástroje např. Bizagi Modeler pro modelování jazyka BPMN nebo instrument Archi pro modelování jazyka ArchiMate. Žádný z těchto nástrojů však

nenabízí možnost propojení více syntaxí, a proto se stávají nevhodnými pro modelování komplexních diagramů propojujících více jazyků. (SPARX SYSTEMS, 2016)

4 Praktická část

Praktická část se zabývá využitím jazyků ArchiMate a BPMN sloužících pro modelování podnikové architektury. Každý z těchto jazyků je trochu odlišný a každý má jiný význam použití, oba však slouží ke grafickému mapování podnikové architektury. Tato práce spojí oba jazyky a využije potřebné elementy diagramy jazyka ArchiMate a orientaci na modelování procesů jazyka BPMN. Ač spojení těchto dvou syntaxí zní velice zajímavě, v praxi se dosud moc nepoužívá. Není to z důvodu nevhodnosti spojení těchto jazyků. Důvodem je, že lidem nejsou k dispozici potřebné materiály, a pokud si nezaplatí nákladné semináře a školení, nemají do zmíněné problematiky taková náhled. Z tohoto důvodu si lidé vybírají pouze jeden jazyk, který je pro ně vhodnější a jeho semináře navštěvují. Někdy však ani tyto semináře nepřinesou uživateli tolik, aby mohl hned sám zvládnout zmodelovat strukturu složitější firmy. Veřejných dokumentů moc není a ty co se vyskytují, jsou pro nezasvěceného uživatele složité a zabere mu mnoho času je nastudovat.

V případě, že začátečník modeluje jednoduchý či složitější systém, pomocí chaotické metodiky, může dojít do bodu, kdy jeho modely budou pro praxi nepoužitelnou a neuchopitelnou věcí. V takovém případě jsou modely uschovány do „šuplíku“ a tím i ztrácí svůj smysl. Proto si tato práce za cíl klade nejen zmapování firemní architektury, ale také vytvoření struktury přehledných složek a modelů, ve kterých bude jednoduché se orientovat. Toto je důležité, protože k modelu nebude přistupovat pouze jeho tvůrce, ale také mnoho dalších lidí, kteří musejí najít to, co hledají. S tímto také souvisí používaná detailnost modelu a použitá odbornost. Jak již bylo zmíněno v teoretické části, modely nesmí být



Obrázek 130 Project Browser (zdroj: vlastní tvorba)

příliš detailní, aby nedošlo k jejich nepřehlednosti, ale také požitá přílišná odbornost, může způsobit nepřehlednost modelu, neboť někteří jeho uživatelé daným výrazům nemusí porozumět.

Nemodelují se procesy firmy, ale procesy našeho produktu. Jelikož jsme vývojáři ERP systému, tak náš produkt obsahuje mnoho procesů, kterými se řídí poté naši zákazníci. V první řadě dojde k vytvoření diagramů této firmy, aby mohl být systém předveden zákazníkovi a ten měl lepší představu o nabízených možnostech. V dalších krocích dojde k zmapování firemní struktury přímo u klienta, aby podle jeho požadavků a procesů mu mohl být vytvořen systém přímo na míru.

4.1 Přestavení společnosti

Společnost se zabývá vývojem informačních systémů pro středně velké firmy. Kromě vytváření informačních systémů jsou v nabídce pro zákazníky poskytovány také podpůrné služby, mezi které patří cloud computing, údržba a případné řešení reklamací nebo dodatečnou implementaci jednotlivých modulů. Klienty společnosti jsou firmy nacházející se především ve výrobním průmyslu. Zákazník si může systém zakoupit samotný a nechat si ho naimplementovat na své servery, nebo si může systém zakoupit pouze jako službu a přistupovat k systému vzdáleně a ten zůstane uložen na serverech vývojářské firmy.

4.2 Modelování procesů

Je zapotřebí znát firemní procesy při vytváření systému? Při tvorbě informačního systému je důležité, aby byl k dispozici dotyčný, který tento systém dobře zná. Ale jak docílit, aby systém odpovídal přesným požadavkům zákazníka a tento zákazník, po předání softwaru netvrdil, že si daný systém představoval jinak? Na toto je jednoduchá odpověď. Pokud firma, která vytváří pro zákazníka systém, zná jeho procesy, je mnohem jednodušší stanovit požadavky a systém podle těchto požadavků a procesů vytvořit přesně na míru. Z tohoto je však patrné, že k takovému vývoji nestačí pouze programátor, ale je zapotřebí celá řada pracovníků, a především ochota zákazníka spolupracovat. Zmapované procesy pak přinášejí dvojitý užitek. Nejen, že pomohou dodavateli při pochopení firmy a vyvíjení systému, ale také umožní konkrétní firmě přehledně zpracovat své procesy pro

budoucí použití. V tomto kroku však nastává problém, protože spousta zákazníků a odborných pracovníků, kteří znají své firemní procesy, mají dost své práce a představa další činnosti je od mapování procesů odrazuje. Dokonce se najdou tací zákazníci, kteří nový systém ani neviděli a stačí jim pouze odpověď, jestli systém splní jejich požadavky a v případě, že ano, tak systém berou tak jak jim je nabídnut. Dalším z příkladů odmítání grafického mapování je ten, že mají své textové dokumentace a ty jim plně vyhovují. Takové dokumentace bývají však nepřehledné a s rozvíjející firmou tak velké, že přestávají být udržitelné. Proto v mnoha firmách dochází ke stavu, kdy mají sice textové dokumentace, ale reálně vykonávané procesy se od této dokumentace odlišují.

Takoví lidé, kteří odmítají grafické modelování, však často ani nevědí, o co se jedná, nebo prostě odmítají inovace. Avšak spousta z nich změní názor ve chvíli, kdy pochopí, o co se jedná. A proto i firma musí jít s dobou a své zákazníky do dané problematiky zasvětit hned na začátku, a to nejlépe tak, že bude mít zmapované své procesy nebo procesy systému, který nabízí a tento systém bude zákazníkovi prezentovat touto grafickou formou.

Takováto grafická forma může také napomáhat firmě s vysokou fluktuací zaměstnanců. Pokud má firma dobře zmapované své procesy, může je použít k zaškolení nových pracovníků, kteří si mohou svou náplň nastudovat pomocí zmíněné grafické notace. To snižují zatížení odborného pracovníka.

V dnešní době se aktuální situace velmi rychle mění a vyvíjí. Počítač, který si člověk koupí, za pár let začne být nedostačující jak výkonem, tak i funkcemi, které nabízí. Co se týče mobilních telefonů, je toto stárnutí ještě daleko rychlejší. S tím, jak se svět vyvíjí, je kladen neustále větší požadavek na výpočetní výkon. Množství zpracovávaných dat stále narůstá. Kdo by si myslel, že u softwaru je to jinak, tak by se mýlil. Systémy se vyvíjí stejně jako hardware podle potřeb doby a žádná firma nemůže jeden software používat beze změny věčně.

Nasazení nového nebo úprava stávajícího systému, tím spíš, jedná-li se o zakázkový systém, není však tak jednoduchá jako výměna některého z počítačů. Firma postupem let dosáhla určitých změn ve svých procesech a ve chvíli, kdy chce nový systém, musí dodávající podnik projít celým procesem poznávání firmy, mapování procesů a požadavků znovu, jako kdyby zde byla úplně poprvé. Tomuto se však dá

velice snadno předejít. Využije-li se při mapování firemních procesů některá ze syntaxí, jako jsou např. ArchiMate, BPMN nebo UML, namísto zdlouhavých a náročných textových dokumentací, je pak mnohem snazší se v dané firmě zorientovat. Firma vyvíjející systém si tyto procesy vezme a velmi rychle zjistí, jak to v dané firmě funguje a jaké všechny služby současný systém nabízí. Důležité je, aby vytvořené podklady se měnily s firmou a nezůstaly po vytvoření založené někde v „šuplíku“. Pokud jsou grafické dokumentace stále aktuální, není pak problém, ať už pro stávajícího dodavatele nebo nového, se ve firemních procesech vyznat a nasadit nový systém v daleko kratším čase.

Další výhodou je vysoká variabilita při změnách nebo snadnost hledání incidentu, který nastal v některém z procesů. V zmapovaných grafických procesech se příslušný pracovník velmi rychle zorientuje a může tak pohotově provést změny nebo pomoci najít problém. V případě, že firma používá textové nebo dokonce, nepoužívá žádné dokumentace, nalezení případného problému je velice složité a ve výsledku zabere mnohem více času nežli samotné grafické mapování. Nejhorší variantou je, když firma své problémy neřeší vůbec, nebo je prostě odkládá na později. Z banálních problémů pak mohou vyvstat značné problémy, které naruší celý firemní proces. I přes tyto výhody přináší grafické mapování některé nevýhody. A to z pohledu pracnosti, kdy by měl být vybrán jeden pracovník, který by se o dané procesy staral. A postupně upravovat tak, jak se vyvíjí firma, aby modely byli stále aktuální.

4.3 Postup tvorby grafického modelování

V této kapitole bude probráno, jak postupovat při vytváření grafických modelů pomocí již zmiňovaných dvou syntaxí. Když se dodavatel uchází o zákazníka ve výběrovém řízení, může díky grafickým modelům prezentovat svůj produkt na reálných procesech, a tak i klientovi přiblížit všechny možnosti a funkce celého systému. V případě, kdy již má i zákazník zmodelované své procesy, může dojít k porovnání obou procesních map a tak i určení, do jaké míry se musí dodavatelský systém upravit, tak aby vyhovoval všem zákaznickým potřebám. Z tohoto důvodu je na začátku nutné popsat dodavatelskou a zákaznickou procesní mapu.

V dalším kroku budou v procesní mapě zákazníka identifikovány a zkonkretizovány jeho požadavky, které vychází ze zmapovaných cílů.

4.3.1 Organizační struktura obchodního modelu

Jako první je vhodné si znázornit jednoduchou organizační strukturu, tedy především obchodní model, který bude zákazníkovi znázorňovat produkty a služby, které mohou být vývojářskou firmou nabídnuty. V prvním kroku je vhodné znázornit aktéra jako společnost a k němu přiřadit role, které budou se zákazníkem komunikovat. Na tyto role by měly být napojeny služby, které zákazníkovi budou v rámci zamýšleného produktu nabízeny, až z těchto služeb vznikne zmíněný produkt. Tento diagram obchodního modelu je vhodný, aby zákazník viděl, co vše mu může být nabízeno za služby a firma měla podklady na prezentaci pro zákazníka, a především strukturu rolí, služeb a produktů, které může zákazníkovi nabídnout.

4.3.2 Tvorba byznys vrstvy

4.3.2.1 Modelování byznys procesů

Pro modelování firemních procesů, je možné použít obě ze zmiňovaných syntaxí. Pro jednoduché procesy a vytvoření komplexního modelu postačí jazyk ArchiMate. V případě, že je potřeba vytvářet složitější strukturu procesů, je zde vhodné využít spíše jazyk BPMN.

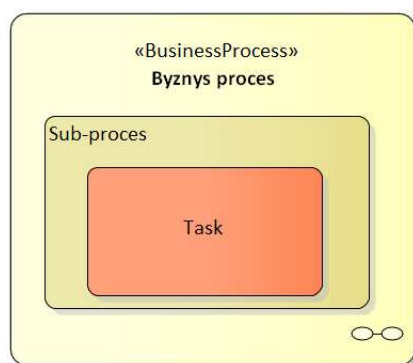
Při modelování procesů se nezačíná vytvářet hned z počátku tok procesů, ale začíná se vždy od hlavních (páteřních) procesů. Tyto páteřní procesy by se měly dále dělit do tří kategorií, kterými jsou: hlavní procesy, řídicí procesy a podpůrné procesy. Hlavní procesy jsou ty procesy, které firmě přinášejí zisk a velmi často oddělují jednotlivé úseky systému, které se mohou prodávat samostatně (např. výroba, prodej, expedice). Druhou skupinou jsou řídicí procesy, jež řídí jednotlivé části, aby došlo k udržení konzistence a logiky ostatních prováděných procesů v organizaci. Tyto procesy jsou realizovány managementem společnosti a samy o sobě nepřinášejí firmě žádný zisk (např. personální zdroje). Třetí skupinou jsou podpůrné procesy, jejichž cílem je zajistit fungování hlavních procesů a organizačních procesů (např. interní audity). Z důvodu rozsahu a implementace

pouze úseku výroby bude v detailnějších modelech rozebrán pouze tento byznys proces.

4.3.2.2 Sub-proces a Task

Z byznys procesů dochází k rozpadu na detailnější sub-procesy a atomické aktivity (tasky). Při vytváření takovéto úrovně, nemusí být ještě modelován tok, protože některé aktivity jsou důležité pro znázornění hierarchie a pochopení firemní firemních procesů a až poté je možné modelovat tok. Důležité je, pokud dochází k rozpadu byznys procesu na sub-procesy nebo sub-procesu na tasky, musí vnořená podmnožina vždy obsahovat, alespoň dvě aktivity, jinak daný rozpad a přílišné detailní modelování nemá smysl.

Zmíněné tasky (atomické aktivity) již není možné dále dělit ani rozpadat na detailnější aktivity. Mnohdy tyto aktivity nejsou ani důležité pro samotný systém, avšak pokud by modelovány nebyly, mohlo by dojít k nejasnému pochopení firmy a s tím spojené špatné optimalizaci nebo tvorbě samotného systému.



Obrázek 131 Rozpad při tvorbě procesů (zdroj: vlastní tvorba)

4.3.2.3 Jak správně vytvářet Workflow

Modelovací jazyky nemají přesně definovaná pravidla, jak se musí postupovat při jejich vytváření a jak mají vypadat výsledné modely. Je to způsobeno značnou otevřeností syntaxí, která dovoluje uživateli využít značnou míru kreativity a vytvořit tak diagramy podle své potřeby. Nesmí však při tom porušit pravidla syntaxe, kterou využívá. Přestože existují určité best practices (nejlepší metody) jak modelovat, diagramy, které jsou vytvářeny, nejsou jednotné. Proto tato práce se

snaží vytvořit jednotný postup, aby při komunikaci mezi pracovníky firmy nebo se zákazníkem nedocházelo k nejasnostem kvůli velké odlišnosti modelů.

Rozložení diagramu

Pro účel vytváření dokumentací z grafických modelů je důležité, aby i samotné rozložení modelu dodržovalo určitá pravidla. Základem je, že modelovaný diagram by se měl vejít vždy na stránku A4. Toto pravidlo je zavedeno z důvodu potřeby převádět dokumentaci do grafické podoby a nedodržení této velikosti vede k nerovnoměrnému rozdělení diagramu na více papírů. Nástroj používaný v této práci napomáhá autorovi vyznačit okraje stránky. Dalším pravidlem je, že diagramy by se měly modelovat vždy od shora dolů a zároveň zleva doprava.

Popis elementů

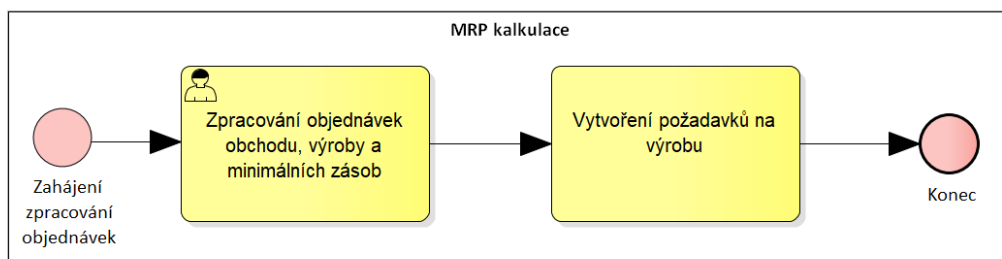
Při vytváření elementů, musí dojít k jejich správnému a vystiženému popisu tak, jak je uvedeno v teoretické části. K popisu by mělo docházet vždy při tvorbě samotného elementu, a pokud je jeho význam složitější, měl by být rozepsán více v poznámce nebo formou textové anotace, tak aby i nezainteresovaný čtenář rychle pochopil, co daný element znamená. Výjimku tvoří události, jejich popis je doporučený, avšak jeho použití je povinné až ve chvíli, kdy se v jednom diagramu nachází více elementů stejného typu (např. start event)

Happy Path

Jak uvádí zdroj (Bizagi, c2002-2018) proces by měl být vytvářen metodou „Happy Path“. Tato metoda naznačuje, že proces by měl být vytvářen jako posloupnost aktivit, které na sebe navazují bez nutnosti řešení podmínek a výjimek. Tato metoda se používá, aby došlo k jasnému vytvoření hlavního toku, neboť pokud by docházelo k přímému modelování podmínek a vedlejších toků, vedlo by to k narušení koncentrace od hlavního toku. A to by vedlo k nárůstu složitosti modelování a také k nepřesnosti modelu. Výhodou tohoto postupu je, že není nutné začínat modelovat vždy od začátku, ale v případě, že zákazník zná konec svého procesu, může se využít zpětného modelování a k vytvoření happy path se může dojít od konce.

Při modelování procesů se nesmí zapomínat na to, že proces a také první aktivita jsou vždy zahajovány událostí. To znamená, že každý proces musí obsahovat startující a ukončující událost.

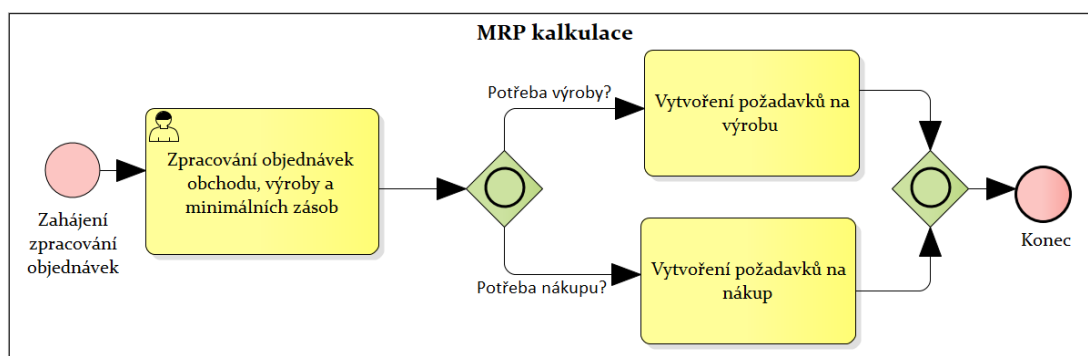
Po vymodelování, i když se jedná pouze o happy path, by mělo dojít k provedení kontroly, zda procesy jsou zmodelovány správně a model odpovídá realitě.



Obrázek 132 MRP kalkulace dle Happy path (zdroj: vlastní tvorba)

Doplnění diagramu o alternativní cesty a procesy

V dalším kroku je provedeno rozšíření stávajícího diagramu o podmínky a možné alternativní cesty, které mohou při procesu nastat. V tomto kroku je důležité dbát na to, aby byly zmodelovány jen významné procesy a nezacházelo se do přílišných detailů, aby model zůstal přehledný. Při modelování alternativních cest se doporučuje používat základní (exklusive) gateway, který je jednoduchý na modelování i pochopení. Ostatní gateway, popřípadě podmínka formou conditional sequence flow, se doporučuje používat pouze v případech, kdy není možné použít základní gateway.



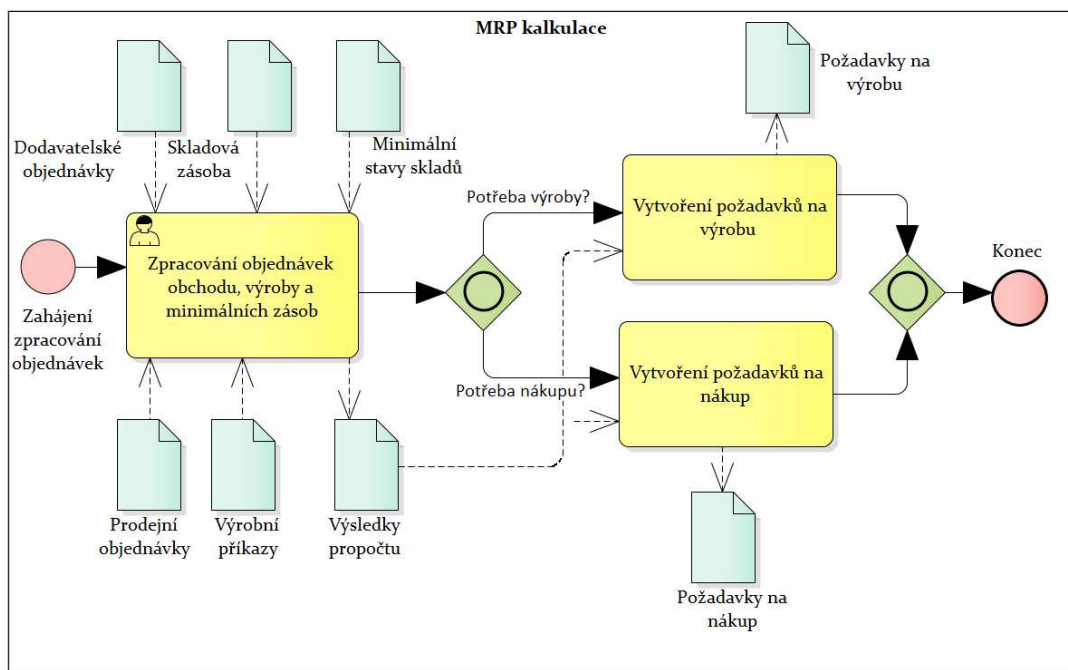
Obrázek 133 Rozšíření MRP kalkulace (zdroj: vlastní tvorba)

Doplnění diagramu o datové objekty

Poté co je celý model hotov, je možné do diagramu vložit datové objekty, které napomáhají znázornit datový tok v procesech, popřípadě, jaká data proces využívá nebo vytváří. Oba v této práci z používaných jazyků nabízejí možnost modelování datových objektů. Je vhodné se dopředu domluvit a používat pouze jeden element

z obou syntaxí, aby nedocházelo k matení čtenáře. Oba datové elementy mohou být sice používány totožně, jejich grafické znázornění je však odlišné a mohlo by čtenáře mást. K připojení datových objektů k aktivitám je vhodné používat vazbu data association, která pomocí šipky znázorní směr toku dat, tedy jestli data do procesu vstupují nebo z něho vystupují. Jiný postup z důvodu jednotnosti a přehlednosti není doporučován.

Úskalím přidávání datových objektů pomocí zmíněné vazby je, že jazyk ArchiMate obsahuje stejně vypadající vazbu „Influence“, která však má jiný význam nežli zmíněná vazba „Data Association“. U vazby influence se však používají matematické funkce „+“ a „-“, které značí, jestli vazba pozitivně nebo negativně ovlivňuje následující element a také by se tato vazba měla používat především v diagramu cílů. Tento drobný rozdíl napomůže čtenáři rozlišit mezi zmíněnými vazbami.



Obrázek 134 MRP kalkule rozšíření o datové objekty (zdroj: vlastní tvorba)

4.3.3 Tvorba cílů

Diagram cílů je vhodné vytvářet, aby bylo možné vidět směr, jakým se firma ubírá a jaké má požadavky na nový systém. Tento diagram napomůže vývojářské firmě a zákaznickovy orientovat se stejným směrem právě při tvorbě nového systému. V první řadě se do diagramu vkládají strategické cíle, což jsou hlavní a dlouhodobé

cíle společnosti. Jednotlivé strategické cíle mohou na sebe navazovat, což může být znázorněno určitými vazbami. Na strategické cíle mohou být stejným způsobem napojeny integrované cíle, jež mají kratší dobu trvání a měly by konkrétněji vystihovat strategický cíl, ze kterého vychází. Poté jsou na tyto cíle napojeny požadavky a případná omezení, které ovlivní dosažení vybraných cílů.

4.3.4 Tvorba aplikační vrstvy

V tomto diagramu je zmodelován samotný systém a jeho části, aby mohl být systém a především služby, které nabízí, představen zákazníkovi. Při vytváření této vrstvy je vhodné začít vymodelováním konkrétního systému a popřípadě i jeho moduly. V dalším kroku jsou k systému připojeny aplikační služby, které jsou systémem nabízeny podobně jako ve struktuře obchodního modelu opsaného výše. V posledním kroku se pak namodeluje propojení na ostatní systémy, které firma také nabízí nebo napojení na systémy, které zákazníci často poptávají, aby z diagramu bylo patrné, že systém je variabilní a umožní používání i stávajících systémů.

4.3.5 Tvorba technologické vrstvy

V této části jsou modelovány stroje, servery a ostatní hardware nebo náčiní používané ve firmě. Model technologické vrstvy by neměl být složitý, měl by pouze odrážet jednoduchou strukturu firmy, do které bude ERP systém zasahovat, aby mohl být systém uzpůsoben počtu zařízení, se kterými bude pracovat a také, aby mohlo být vybudováno dostatečné zabezpečení v případě, že systémy spolu komunikují mimo intranet.

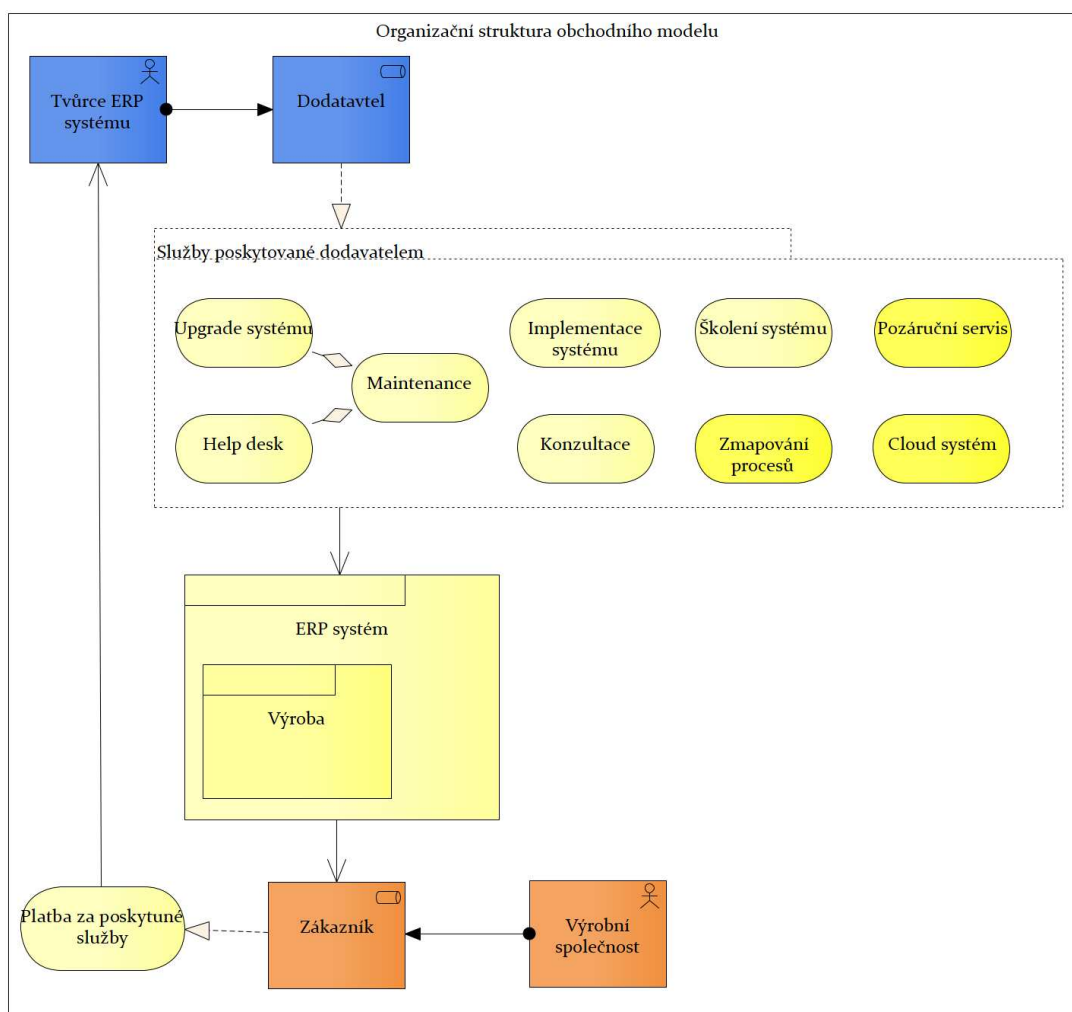
4.4 Výroba – grafické mapování

Pro praktické znázornění výše zmiňovaného postupu byl vybrán vzorový příklad výroby. Touto částí systému se budou zabývat všechny níže znázorněné modely. Nejdříve budou vytvořeny a popsány modely firmy poskytující ERP systém a v dalším kroku budou vytvořeny modely zákazníka, aby dle těchto diagramů mohl být vytvořen ERP systém. Součástí diagramů budou popisky, které čtenáři vysvětlí, co daný diagram znázorňuje. Z vytvořených modelů může čtenář získat

nový pohled na propojení syntaxí ArchiMate a BPMN, ale především jak správně postupovat při popisování firmy pomocí grafických notací.

4.4.1 Organizační struktura obchodního modelu

Tvůrce ERP systému, který stojí v tomto modelu v roli dodavatele, realizuje veškeré služby, které jsou znázorněny v rámečku služeb jim poskytovaných. Služby se skládají z hlavních služeb, které jsou součástí zakoupeného systému, jako je Maintenance, Implementace systému, konzultace a školení systému. Maintenance je služba s roční licenci, kdy dodavatelská firma, poskytuje bezplatnou údržbu, zlepšování výkonu nebo modifikaci atributů. Pod touto službou je možné vidět upgrade systému a help desk, na který se zákazník v rámci této služby může kdykoli obrátit. Další službou je samozřejmá implementace systému, se kterou také souvisí zkušební testovací provoz. Konzultace se zákazníkem v průběhu vývoje systému a zaškolení všech pracovníků, kteří budou systém využívat. Zbylé služby



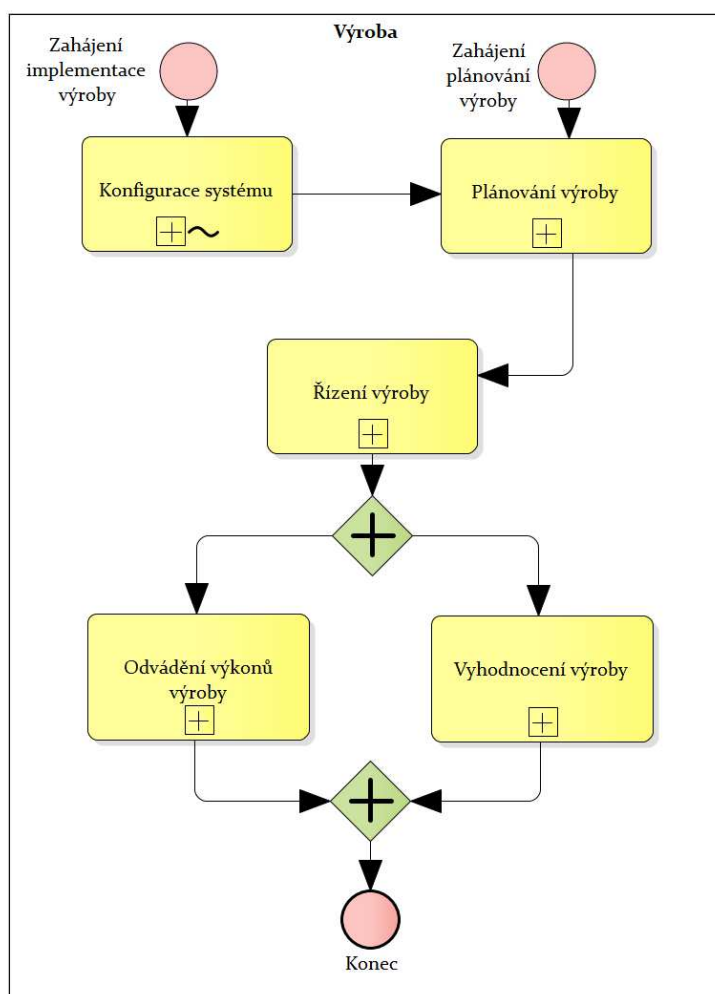
Obrázek 135 Organizační struktura obchodního modelu (zdroj: vlastní tvorba)

jsou příplatkové a zákazník si je může kdykoli přiojednat. Tyto všechny služby jsou zákaznickovy zprostředkovány prostřednictvím nabízeného ERP systému. Výrobní společnost stojí v tomto modelu v roli zákazníka a v případě využití některých ze služeb realizuje dodavateli platby za tyto služby.

4.4.2 Modelování byznys procesů ERP systému

Hlavní proces

Z důvodu popisování pouze jednoho modulu ERP systému nezačíná model nejvyšší úrovní, tj. byznys procesy, ale je znázorněn hlavní proces modulu výroby. V tomto procesu nejsou také vytvářeny veškeré aktivity, které mohou být prováděny ve výrobním oddělení, ale modelují se zde pouze systémové aktivity, které jsou důležité pro vývojovou společnost a také pro prezentaci zákazníkovi.

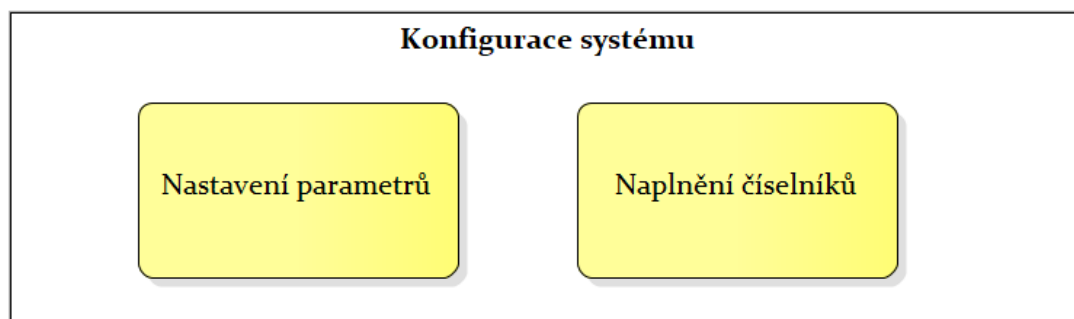


Obrázek 136 Hlavní proces modulu výroba (zdroj: vlastní tvorba)

Diagram popisuje vrchní pohled celého výrobního procesu od nastavení systému, přes plánování výroby až po odevzdání výrobku. Z obrázku je patrné, že výroba může být zahájen dvěma typy událostí. Je-li potřeba nejdříve provést konfiguraci systému, proces je spuštěn první událostí Zahájení implementace výroby. Samotná aktivita konfigurace systému je typu Ad-hoc, to znamená, že všechny vnořené aktivity mohou být spuštěny v libovolném pořadí, avšak nikdy nesmí probíhat dvě současně. Po dokončení konfigurace systému tok procesu pokračuje do aktivity Plánování výroby. Tato aktivita může být také zahájena druhou událostí procesu Zahájení plánované výroby. V takovém případě nedochází ke konfiguraci systému, ale proces začíná rovnou v druhé aktivitě, kde se začíná plánovat výroba. Po této aktivitě tok pokračuje do Řízení výroby, kde dojde k vytištění dokumentace obsahující samotné výrobní úkony, výdejků materiálu ze skladu a mzdové lístky. V dalším kroku se tok rozděluje a spouští současně obě aktivity Odvádění výkonů výroby a Vyhodnocení výroby. Proces může být ukončen až ve chvíli, kdy jsou dokončeny obě tyto aktivity.

Konfigurace systému

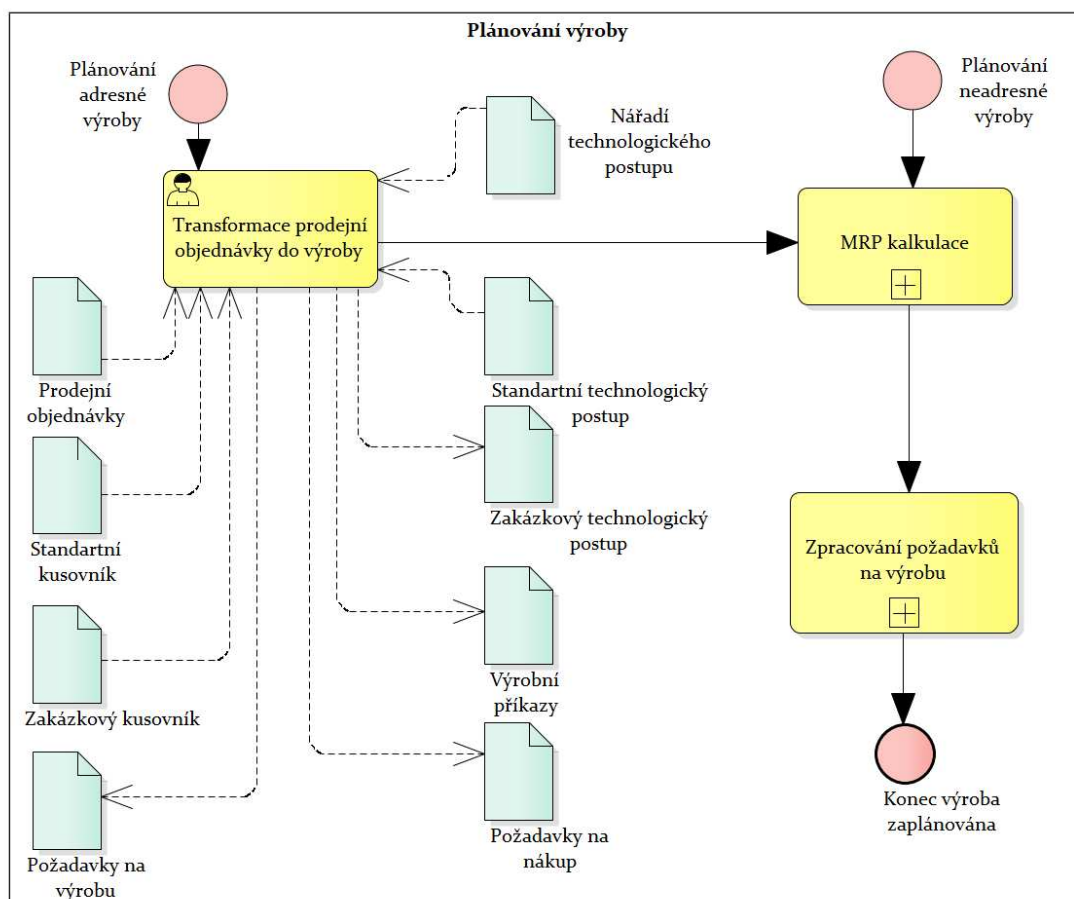
Proces popisující nastavení celého systému, tak aby se číslovaly jednotlivé výrobní zakázky dle určitých pravidel. Aby se jednotlivé výrobky určovaly podle zadaných parametrů. Nastavení jednotlivých nástrojů, operací a všech dalších komponent pro správný chod systému. Aktivity jsou vnořeny v sub-procesu typu Ad-hoc, proto mezi nimi nejsou znázorněny žádné vazby.



Obrázek 137 Konfigurace systému (zdroj: vlastní tvorba)

Plánování výroby

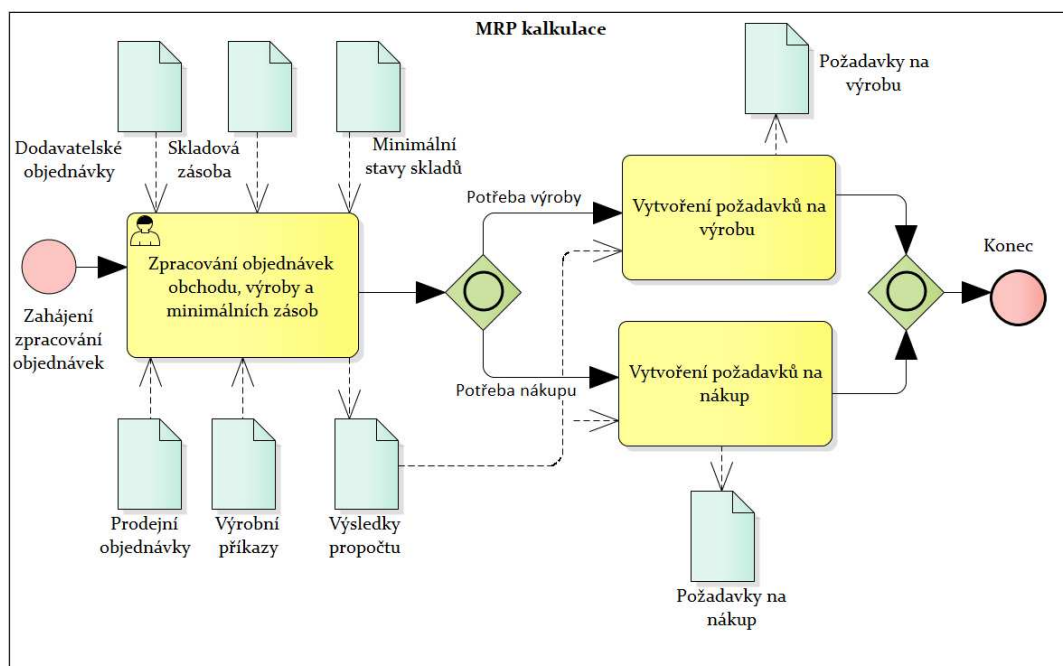
Proces popisující zahájení výrobních úkonů. Samotný proces má dvě startovací události. První událost "Plánování adresné výroby" nastává tehdy, když je potřeba výrobu adresně řídit pro daného odběratele. V takovém případě je již vytvořena výrobní objednávka z již vzniklé prodejní objednávky. V druhém případě vzniká výroba neadresně, tedy budou se vyrábět produkty, které jsou určeny pro doplnění zásob na sklad. Z obou startovacích událostí poté probíhá MRP kalkulace. Po jejím vyhotovení se zpracovávají požadavky na samotnou výrobu. Po zpracování požadavku je produkce naplánována a je připravena k samotné výrobě.



Obrázek 138 Plánování výroby (zdroj: vlastní tvorba)

MRP kalkulace

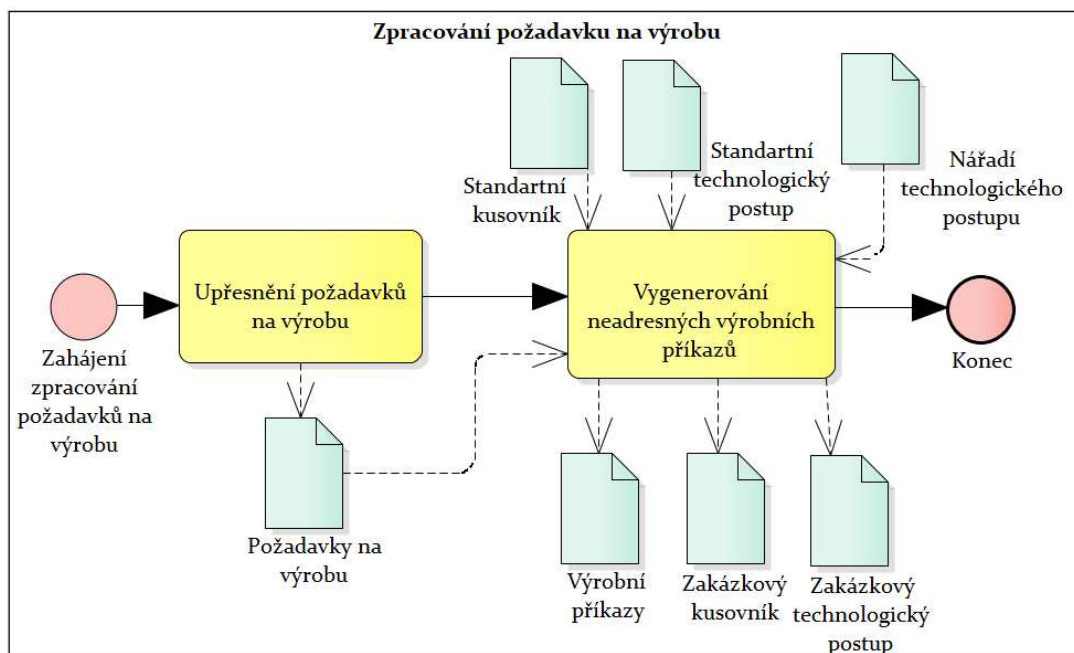
Tento proces popisuje Analýzu výrobních a obchodních objednávek a jejich následné zpracování. V první aktivitě procesu se tyto objednávky zpracují z pohledu zásob, které jsou na skladě. Zjistí se, je-li na skladě dostatek materiálu či polotovaru nebo již vyrobených produktů, tak aby byla pokryta celá objednávka. V tomto bodě může dojít k několika výsledkům. V první řadě dochází ke stavu, kdy je na skladě dostatek materiálu pro výrobu a je tedy vytvořen pouze požadavek na výrobu. V druhém případě může dojít ke stavu, že na skladě již není žádný materiál pro výrobu a musí se v takovém případě zajistit přes oddělení obchodu. Vzniká tak požadavek na nákup, aby byl požadovaný materiál zakoupen. Třetí případ je takový stav, kdy je na skladě pouze část materiálu na výrobu. Pak je nutné vytvořit jak požadavek na výrobu, aby se vyrobily produkty ze zbývajcího materiálu, tak musí být vytvořen i požadavek na nákup, aby se doplnila zásoba požadovaného materiálu.



Obrázek 139 MRP kalkulace (zdroj: vlastní tvorba)

Zpracování požadavku na výrobu

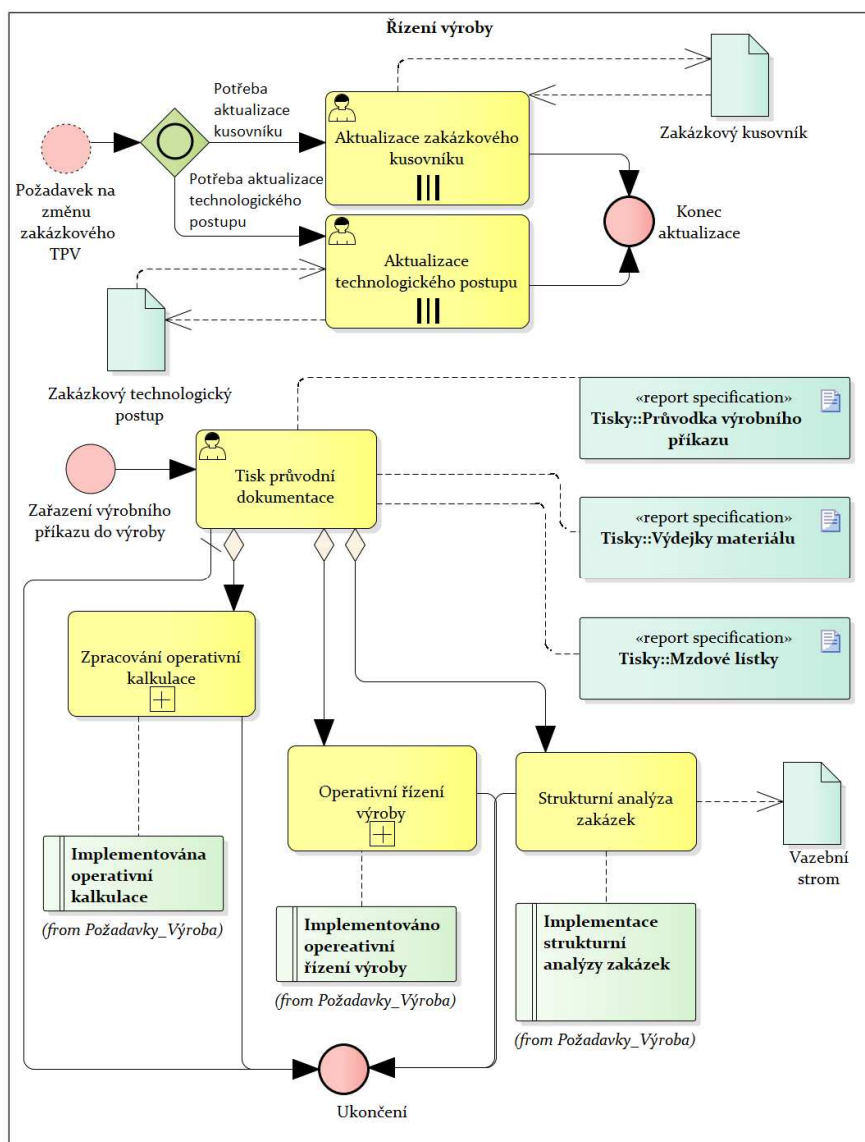
Proces popisuje upřesnění požadavků na výrobu. Jeden požadavek může být precizován z pohledu vnitřních parametrů produktu, výkresové dokumentace či přání zákazníka. Po upřesnění daného požadavku jsou vygenerovány jednotlivé operace, které musí proběhnout, aby byl produkt kompletně vyhotoven.



Obrázek 140 Zpracování požadavku na výrobu (zdroj: vlastní tvorba)

Řízení výroby

Celý proces začíná událostí, kdy je výrobní příkaz zařazen do výroby. Po zařazení příkazu proběhne tisk průvodní dokumentace. Tato dokumentace obsahuje samotné výrobní úkony, výdejku materiálu ze skladu a mzdové lístky. Průvodní výrobní dokumentace řídí jednotlivé výrobní úkony, výdejka materiálu je potřebná pro výdej potřebného materiálu ze skladu a na mzdové lístky se odkazují výrobní pracovníci, kam si zapisují odpracované hodiny strávené na výrobě daného produktu. Tyto dokumenty mohou být v různých formách. V případě, že k výrobě nejsou naimplementovány další komponenty, proces končí.

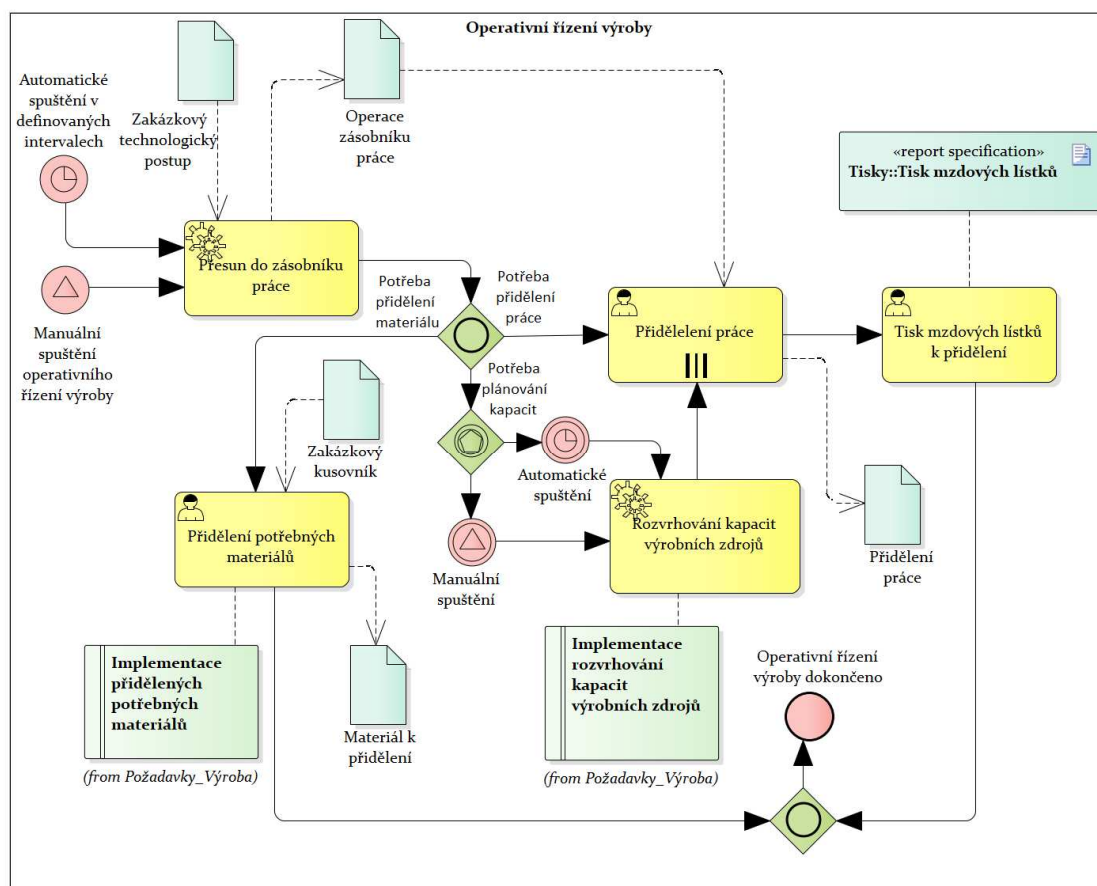


Obrázek 141 Řízení výroby (zdroj: vlastní tvorba)

Pokud je naimplementována ještě operativní kalkulace, proběhne aktivita Zpracování operativní kalkulace. V jiném případě, když je naimplementováno Operativní řízení výroby, uskuteční se stejně zvaná aktivita. Pokud je naimplementována komponenta pro analýzu zakázek, realizuje se stejně zvaná aktivita. Zákazník může mít naimplementovány všechny doplňující funkce, nebo jen některé. V jakémkoliv kroku tohoto procesu, může být spuštěna událost Požadavek na změnu zakázkového TPV. Je to z důvodu, že spouštěcí událost je typu non-interrupting a její spuštění tak neovlivní chod druhého procesu. Podle charakteru situace se nadále aktualizuje buď zakázkový či technologický kusovník nebo mohou být aktualizovány oba.

Operativní řízení výroby

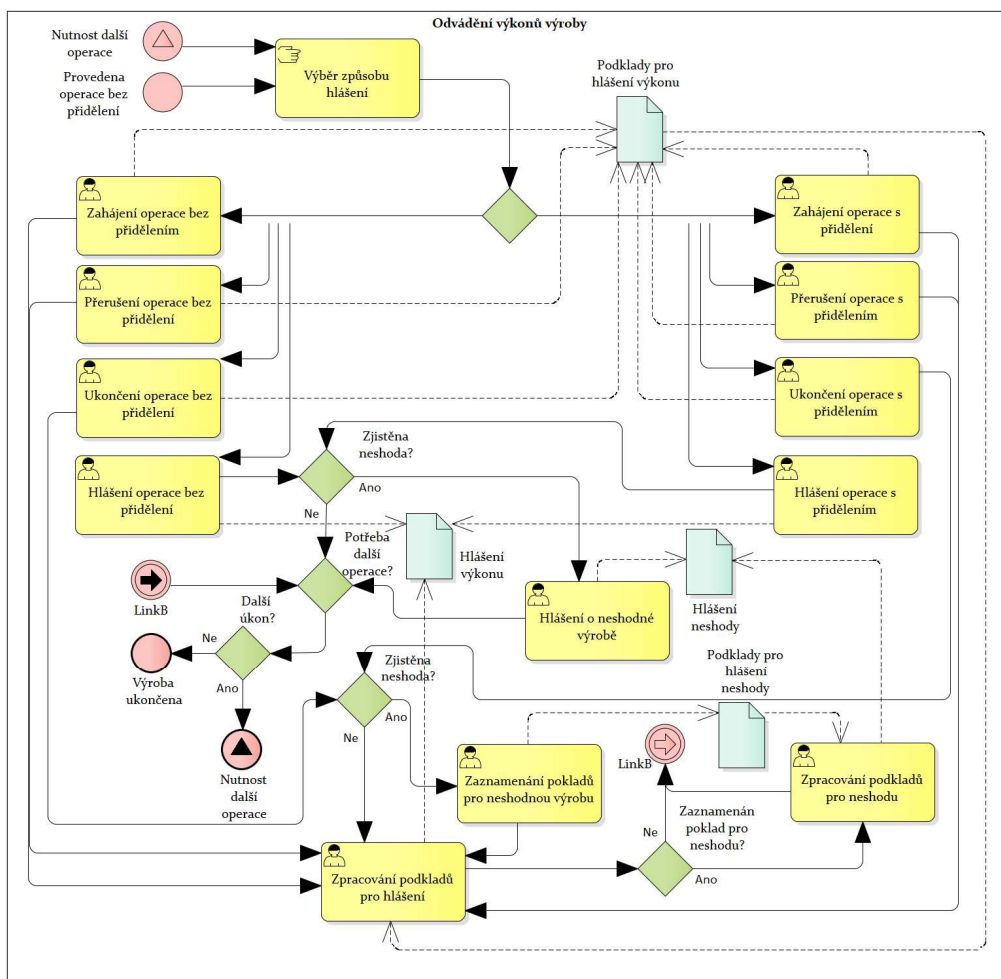
Tento proces je zahájen dvěma událostmi. První událost je automatické spuštění po uplynutí daného časového intervalu. Probíhá tedy automaticky. Druhá událost je manuální spuštění procesu a je určena člověkem. První aktivitou po těchto událostech je Přesun operací do zásobníku práce, kde se určí další chod procesu. Kde může dojít k dalším aktivitám jako přidělení potřebného materiálu na úkon a přidělení práce jednotlivým pracovníkům, kde jsou vytištěny mzdové listy k přidělení. Když je k výrobě naimplementována funkce Rozvrhování kapacit výrobních zdrojů, dochází před přidělením práce ještě k rozvržení kapacit. Zde jsou rozděleny cesty tak, že se rozlišuje, jestli probíhá rozvržení kapacit manuálně či automaticky.



Obrázek 142 Operativní řízení výroby (zdroj: vlastní tvorba)

Odvádění výkonů výroby

Tento proces je zahájen dvěma událostmi. Jednou událostí je zpracování daného úkonu ve výrobě. Druhou událostí je opětovné zahájení dalšího úkonu. Po těchto událostí je provedena aktivita Výběr způsobu hlášení, tedy vybere se, jestli bude daná operace zahájena, přerušena, ukončena či nahlášena. Tyto aktivity jsou buď provedeny s přidělením, nebo bez přidělení. Po výběru a provedení aktivit typu zahájení, přerušeni či ukončení proběhne zpracování podkladu pro hlášení a pokud nebyl nalezen žádný podklad pro neshodu, je celý proces ukončen. Bud' se proces opakuje znovu pro jinou operaci, nebo při ukončení celé výroby proces již opakovan nebude. V případě výběru aktivit typu hlášení proběhne kontrola o neshodě.

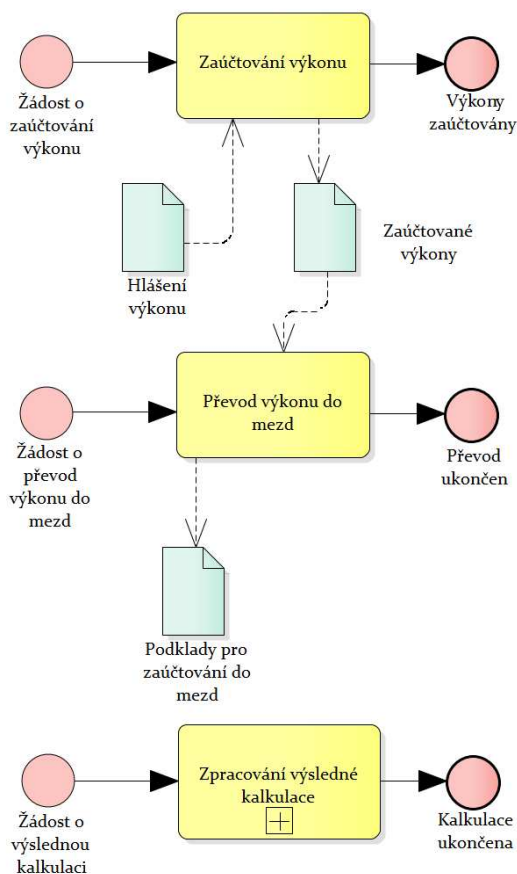


Obrázek 143 Odvádění výkonů výroby (zdroj: vlastní tvorba)

Pokud není nalezena neshoda, proces končí nebo je opakován. Je-li ovšem nalezena chyba musí proběhnout hlášení o neshodné výrobě. Jestliže je vybrána aktivita typu ukončení proběhne kontrola o neshodě a pokud není nalezena, zpracují se podklady pro hlášení a tok pokračuje, jak bylo již zmíněno. Ovšem byla-li nalezena chyba, zaznamenají se podklady pro neshodnou výrobu a zpracují se podklady pro hlášení a pro neshodu. Poté se opět proces buď ukončí nebo znovu probíhá nad jinou operací.

Vyhodnocení výroby

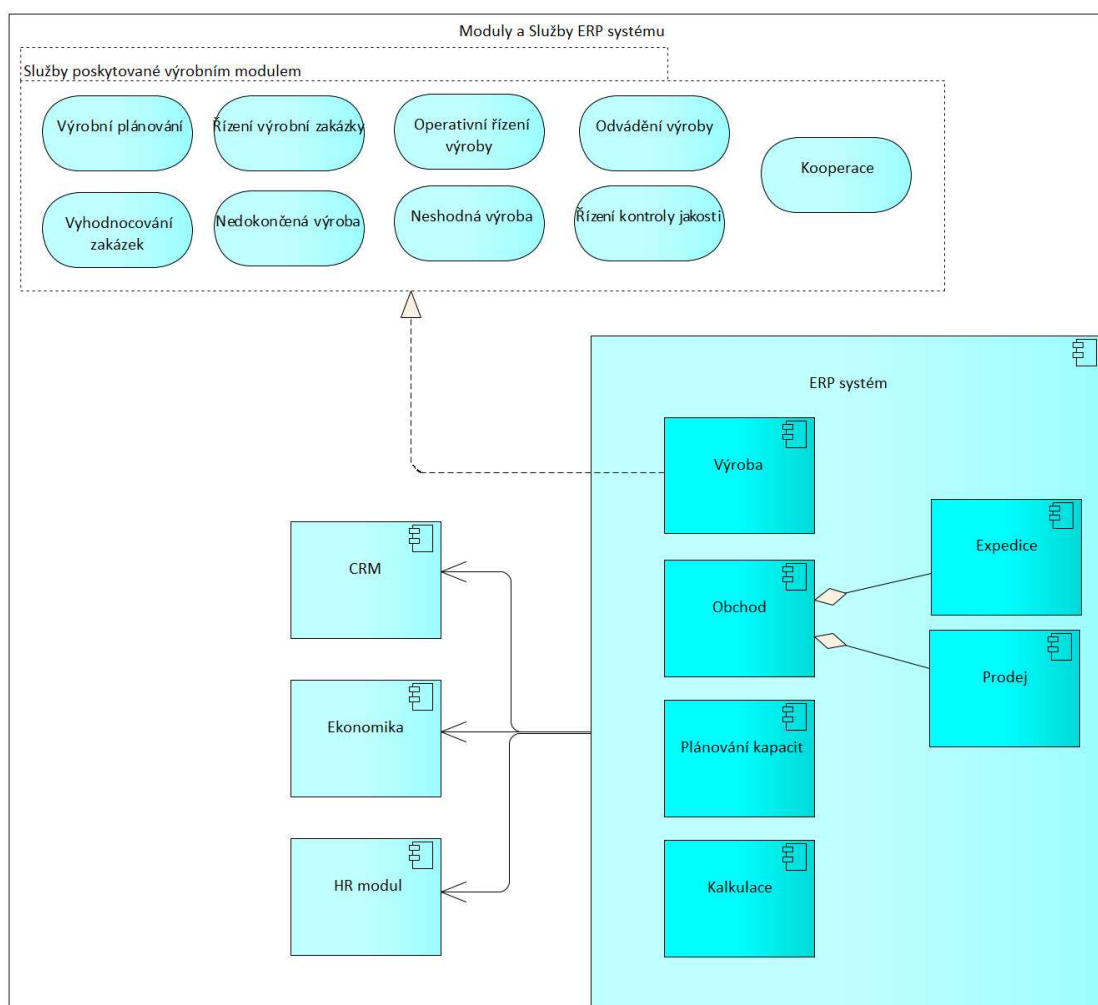
Tento proces je zjednodušen a dán na nejvyšší úroveň, i když je patrné, že všechny jednotlivé toky, by měly být v samostatných sub-procesech. V tomto procesu se vyhotovují tyto činnosti: Zaúčtuje se výkon výroby. Převede se výkon do mezd, tedy přidělí se práce jednotlivých pracovníků do mezd a zpracují se výsledky celé kalkulace. Po vyhotovení všech těchto aktivit je vyhodnocení výroby u konce.



Obrázek 144 Vyhodnocení výroby (zdroj: vlastní tvorba)

4.4.3 Aplikační vrstva

Aplikační vrstva poskytuje globální pohled na nabízený systém a přehled modulů a služeb, které jsou nabízeny. Diagram dále popisuje, že ERP systém se skládá z modulu výroby, obchodu, plánování kapacit, kalkulace, expedice a prodeje, které není možné využít bez zakoupení modulu prodej. V případě, zájmu si může zákazník zakoupit další systémy jako je CRM, Ekonomika nebo HR modul, které je možné propojit s ERP systémem. Komponenta ERP systému, tedy znázorňuje všechny možné moduly, jaké si zákazník může zakoupit a také jaké další systémy a propojení firma nabízí a umožňuje. K modulu výroba jsou napojeny služby, které tyto aplikace zákaznickovy v rámci ERP systému mohou být nabídnuty. Služby, které dodavatel systému znázorňuje jako aplikační, vstupují u zákazníka do procesů jako byznys služby, které napomáhají vykonávat podnikové procesy.



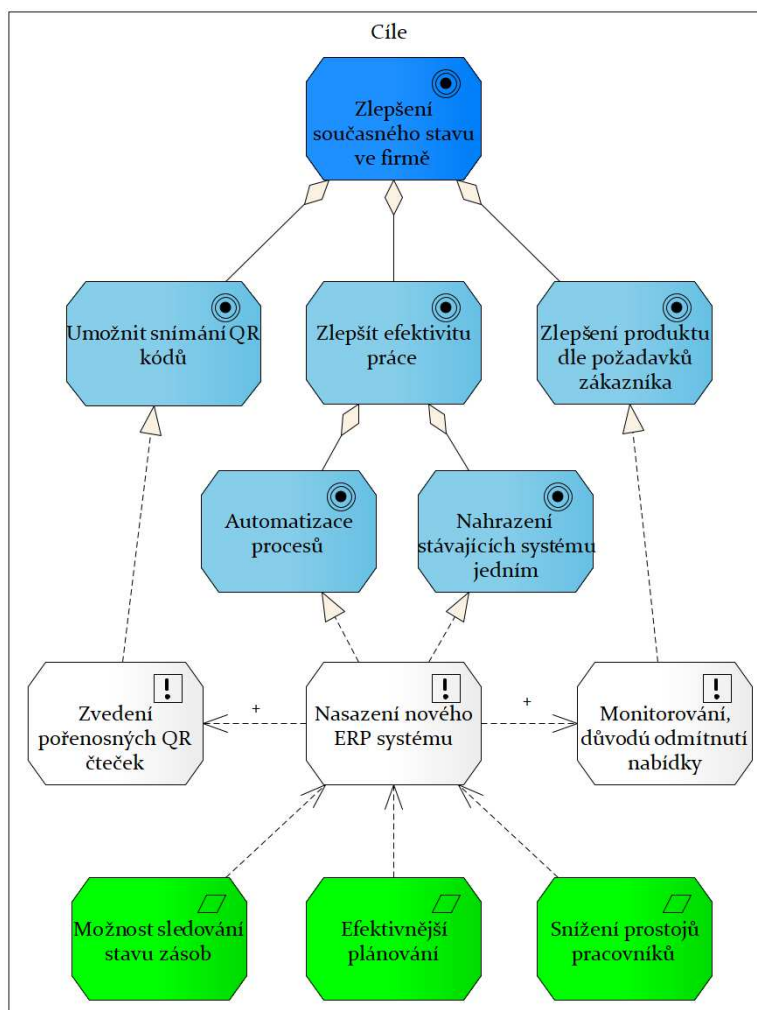
Obrázek 145 Moduly a služby ERP systému (zdroj: vlastní tvorba)

4.5 Modelování u zákazníka

4.5.1 Definování cílů

Při příchodu k zákazníkovi je nejdůležitější zjistit, jaké jsou jeho cíle a důvody, proč chce nahradit stávající nebo zavést nový systém. Tyto cíle a požadavky pomohou nejen modeláři, ale také vývojové firmě zjistit, na jaké části procesů je důležité se zaměřit. Model zachycuje cíle společnosti, kterých má být dosaženo po implementaci nového ERP systému. Strategický cíl nevyšší úrovně se zabývá dosažením zlepšení současného stavu ve firmě. Aby tohoto cíle mohlo být dosaženo, musí být splněny cíle na nižší úrovni, na které se tento cíl rozpadá. Umožnění snímání QR kódu lze zrealizovat zavedením přenosných čteček, které tento kód umožní číst. Zlepšení produktu dle požadavků zákazníka je možné uskutečnit tím, že budou monitorovány všechny neúspěšné nabídky. U těchto nabídek se bude zjišťovat zpětná vazba u zákazníků a následná analýza důvodů odmítnutí nabídky. Cíl zlepšení efektivnosti práce se dále rozpadá na další dva podcíle, kterými je Automatizace procesů a Nahrazení stávajících systémů jedním. Oba tyto podcíle je možné zrealizovat nasazením nového ERP systému, jenž umožní automaticky zpracovávat některé manuální činnosti a také nahradí stávající systémy jedním systémem, což povede k větší stabilitě systému a zvýšení jednoduchosti při školení nových pracovníků. Tento princip Nasazení nového ERP systému je klíčový, protože také pozitivně ovlivňuje (znaménko plus u vazby) Zavedení přenosných QR čteček a Monitorování důvodů odmítnutí nabídky. Na princip Nasazení nového ERP systému jsou také připojeny tři požadavky, které jsou definovány zákazníkem. V případě modelování požadavku v jazyce BPMN může tento model sloužit jako podklady pro jejich tvorbu.

Při pohledu na tento diagram vyvstávají určitá omezení, která by mohla ohrozit realizaci znázorněných cílů, jako je např. dostatečný HW pro implementaci systému. Pokud však zákazník nemá spolu s požadavky i určitá omezení, je lepší tato spíše spekulativní omezení nechat až k případným modelům, kde se bude daná problematika řešit a zde omezení vůbec nemodelovat.



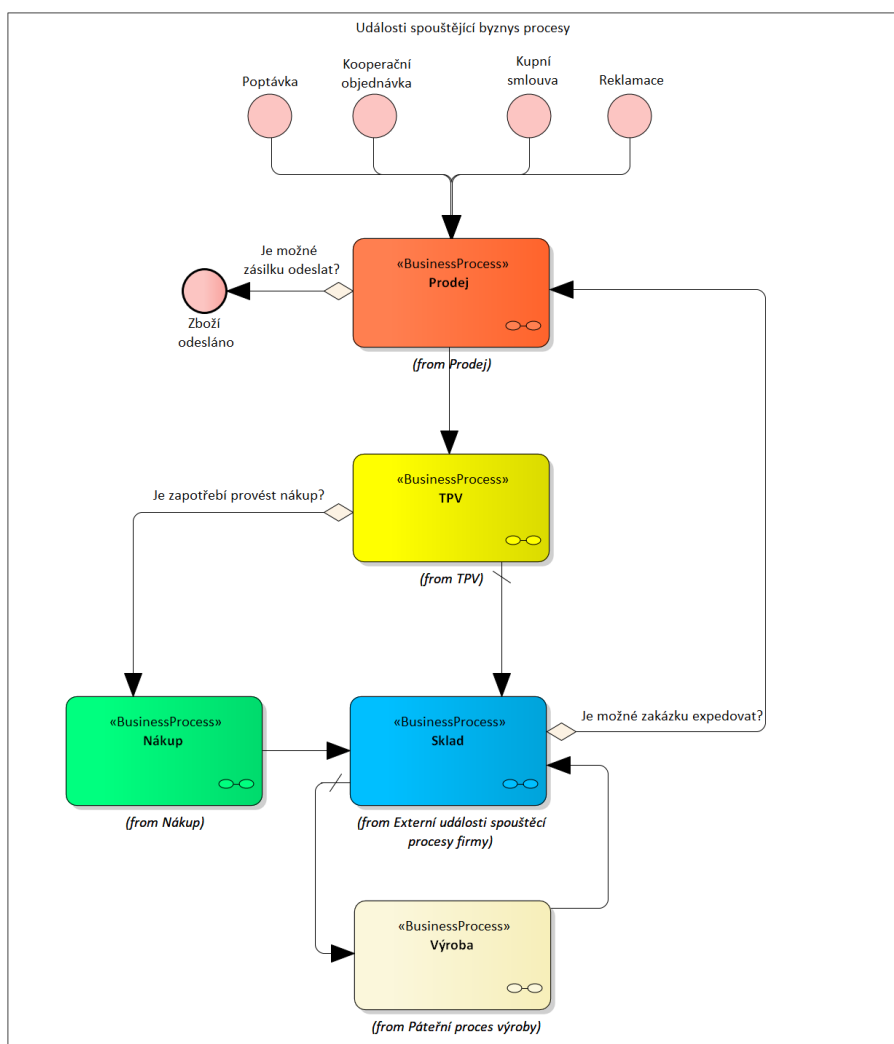
Obrázek 146 Cíle (zdroj: vlastní tvorba)

4.5.2 Modelování současného stavu výrobní firmy

Při mapování procesů u zákazníka je důležité poznat firmu a její okolí, a to i přesto je-li práce cílena pouze na jednu část, neboť veškeré procesy se navzájem prolínají a jeden bez druhého nemůže fungovat. Pro pochopení současných procesů je také nezbytné zmapování i okolních a nesystémových procesů. Jak uvádí Řepa (2008) ve své metodice, nejdříve by měly být zmapovány veškeré události, které přicházejí zvenčí (od zákazníků, dodavatelů atd....) do podniku a spouští hlavní procesy, tedy procesy, které přináší firmě zisk. V dalším kroku se již zmapují ostatní události a jejich posloupnost v rámci firmy. Vhodné je tyto byznys procesy odlišit od sebe různými barvami. V případě, že v detailním modelování vstupují do jednoho procesu aktivity z jiného procesu, tyto aktivity jsou jinak zbarveny a čtenář dokáže snadno rozlišit, do jaké části systému aktivity patří.

Události a byznys procesy výrobní firmy

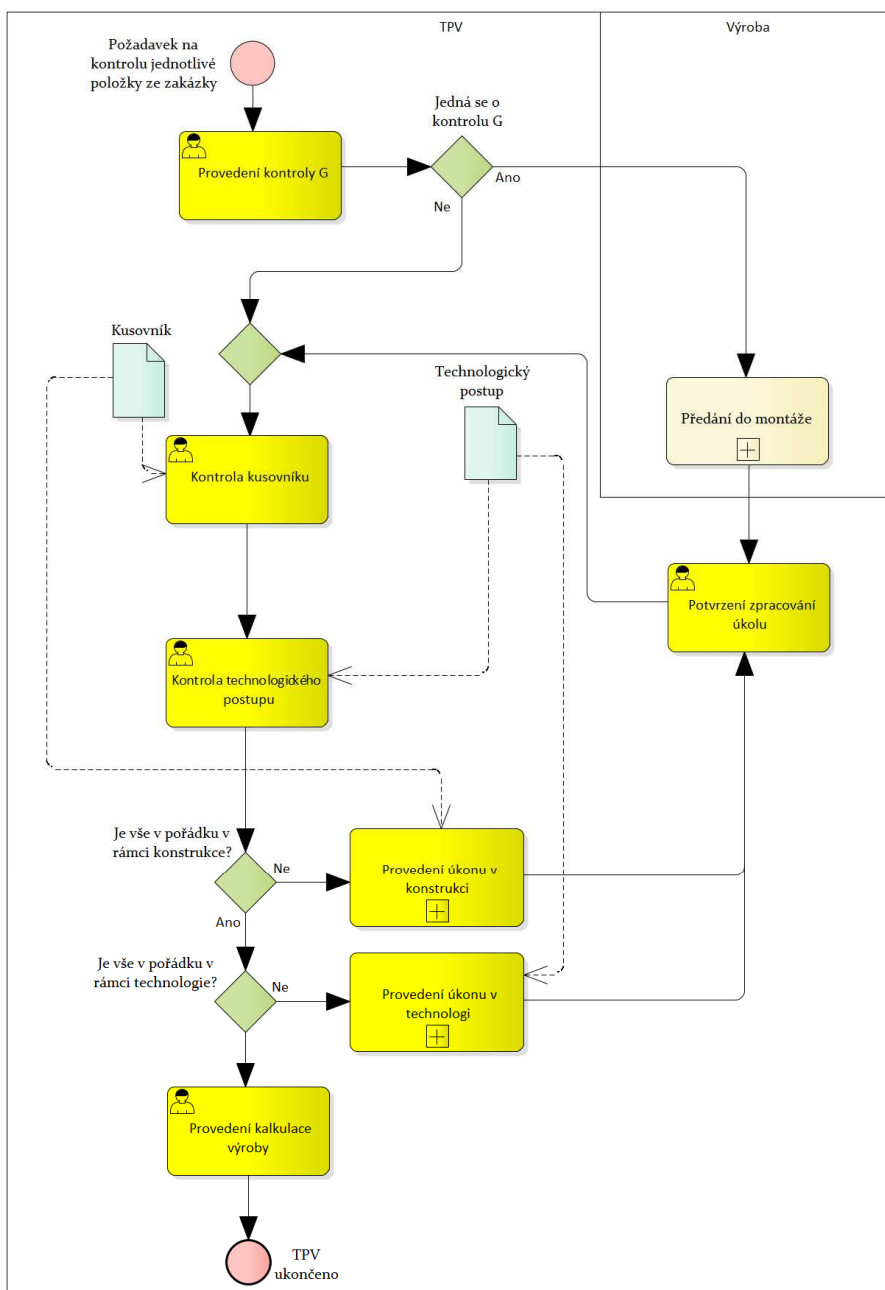
Z níže vloženého diagramu je patrné, že hlavním procesem je prodej, kde veškeré aktivity začínají a také končí. Prodej může být spuštěn, nastane-li některá ze čtyř událostí: Poptávka, Kooperační objednávka, Kupní smlouva, Reklamáce. Z prodeje se tok přesune do procesu technická příprava výroby (TPV), kde po jeho dokončení tok pokračuje dále do skladu a v případě, že je splněna podmínka u sekvencního toku, pokračuje se nákupem. Až poté se jde do skladu. Podobně je to také ve skladu. Je zkontrolována podmínka, jestli je možné zboží expedovat a v případě, že není, proces pokračuje do výroby, kde po jejím dokončení se výrobek vrací zpět na sklad. Vše se opakuje do té doby, dokud není možné expedovat. V prodeji se ověří, zda je možné zásilku odeslat a tím končí celý koloběh jedné objednávky.



Obrázek 147 Události a byznys procesy výrobní firmy (zdroj: vlastní tvorba)

Technologická příprava výroby

TPV není samotným výrobním procesem, ale je s ním úzce spojen a jak je vidět v diagramu níže, objevují se zde výrobní procesy a tato část je nejčastěji součástí výrobního modulu.



Obrázek 148 Technologická příprava výroby (zdroj: vlastní tvorba)

Po dokončení prodeje je spuštěn proces TPV, jak je patrné z diagramu byznys procesů. V okamžiku, kdy je v prodeji vytvořena a vytištěna kupní smlouva, je spuštěn proces TPV. V první řadě je provedena kontrola G. Jedná se o kontrolu, zda

bude vyráběn nový výrobek, nebo byla přijata poptávka na generální úpravu, modifikaci, či např. reklamaci výrobku. V takovém případě se jedná o výrobek G a ten pokračuje do montáže na provedení některých ze zmíněných úkonů. Pokud se nejedná o G výrobek, proces pokračuje ke kontrole kusovníku, kde uživatel v systému musí provést kontrolu všech podsestav, dílů a výchozího materiálu, ze kterých bude vyráběno. Dále uživatel provede kontrolu technologického postupu souběžně s kontrolou kusovníku. Po dokončení obou kontrol tok pokračuje dál do podmínky, jestli je vše v rámci konstrukce v pořádku. V případě, že není, tok pokračuje do konstrukce, kde jsou provedeny konstrukční změny, které musí být potvrzeny a tok se vrací zpět ke kontrole kusovníku a technologického postupu. Tento koloběh je prováděn do té chvíle, než bude vše v rámci konstrukce v pořádku. Poté musí být zjištěno, je-li vše v pořádku v rámci technologie a proces probíhá obdobným způsobem jako u podmínky s konstrukcí. Po úspěšné technologické kontrole je provedena kalkulace a proces TPV je ukončen.

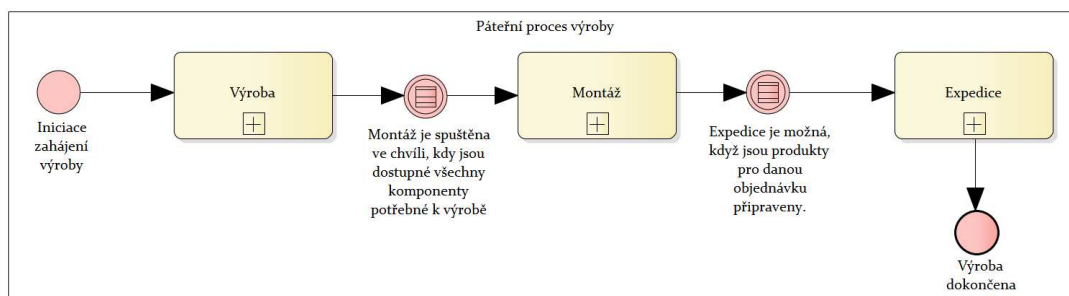
Čtenáři v tuto chvíli může být divné, že autor klade důraz na to, aby při vytváření procesů bylo postupováno vždy ve stylu Happy Path, ale u popisu dodržován není. Tento způsob modelování byl při vytváření dodržen a také se osvědčilo, že v případě odloučení od této cesty dojde ke ztrátě a nejasnosti posloupnosti aktivit ze strany grafického architekta. To je způsobeno dosavadní neznalostí současných procesů, a proto se doporučuje nejdříve udělat hlavní cestu a poté až možné alternativy. Diagram Happy Path ani jeho popis zde při komentování diagramů není nutné používat, neboť by docházelo ke zbytečnému natahování popisu.

Páteří proces výroby

Nastane-li situace, že byznys proces se rozpadá do více úrovní a ne pouze do jedné, jak tomu bylo v případě procesu TPV, je vhodné vytvořit druhou úroveň ze sub-procesu, který budou obsahovat samotný proces a ten se také nůž propadat ještě na nižší úrovně.

Tento diagram je velmi jednoduchý: nejdříve bude spuštěna výroba, poté montáž a finální výrobek bude expedován. Mezi jednotlivými sub-procesy jsou však události obsahující podmínku a bez jejího splnění nemůže tok pokračovat na další proces. Vezme-li se např. podmínková událost mezi výrobou a montáží, je z ní

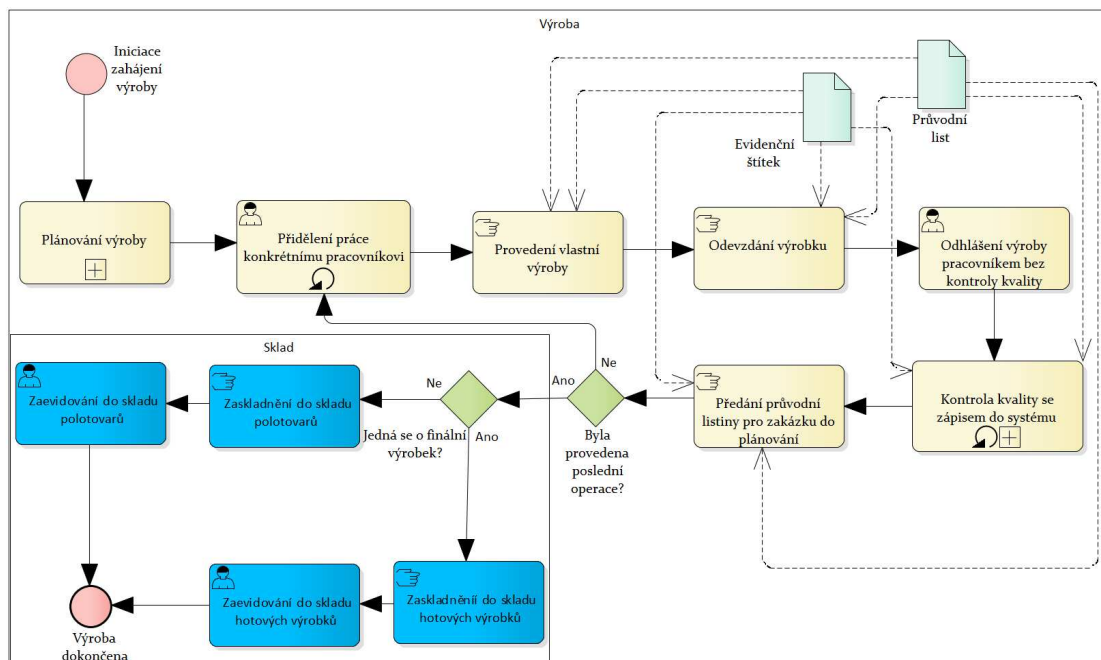
patrné, že montáž nemůže začít, dokud nejsou dostupné všechny díly potřebné k složení finálního výrobku.



Obrázek 149 Páteřní proces výroby (zdroj: vlastní tvorba)

Výroba

Proces výroby začíná jejím naplánováním. Jak již bylo zmíněno výše, tento proces se bude dále rozpadat a bude popsán níže, neboť jeho složitost by mohla vést k velkému diagramu a k jeho nepřehlednosti. Dalším krokem v procesu je přidělení práce konkrétním pracovníkům. V této aktivitě se bude postupně přidělovat práce každému z pracovníků, dokud nebude všechna práce rozdělena. Pracovník si vezme svou přidělenou práci a jde jí vykonat. Po vyhotovení výrobku musí být práce fyzicky odevzdána. Po jejím odevzdání musí pracovník dojít k počítači a zadat, že je práce hotová. V systému v části přidělení práce se zaměstnanec zobrazí s označením k dispozici pro zadání nové práce a říká tím, že současnou práci má již hotovou. V tuto chvíli, ale výrobek ještě není hotov. Nejprve musí projít kontrolou kvality a až poté může být označen za dokončený výrobek. Na tento fakt výrobní pracovník nemusí čekat a může již v tuto chvíli produkovat výrobek další. V dalším kroku dojde k předání průvodní listiny zpět do plánování a pokud se nejednalo o poslední operaci daného produktu, je přidělena práce pracovníkovi. V případě, že se jednalo o poslední operaci v průvodní listině, dojde k zaskladnění a zaevidování výrobku. Předtím však musí být rozhodnuto, jestli se jedná o finální výrobek a půjde na sklad hotových výrobků, nebo o polotovár, který poputuje do jiného skladu.

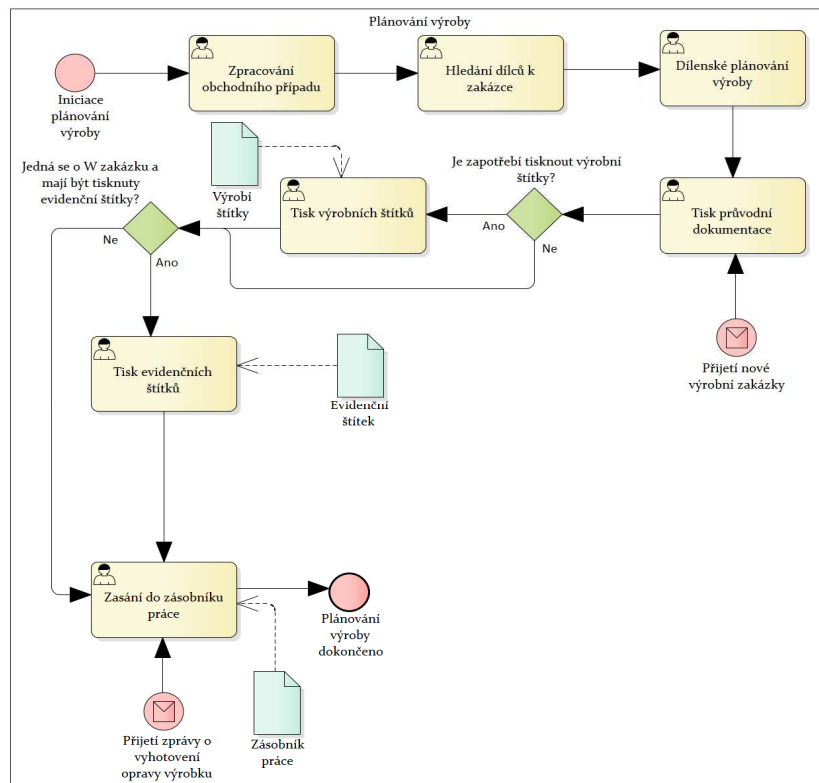


Obrázek 150 Výroba (zdroj: vlastní tvorba)

Plánování výroby

Tento proces probíhá hned na začátku procesu výroby, a tedy i před všemi aktivitami popsanými v předchozím diagramu. Pro obecný pohled na výrobu, však tento detailní proces plánování není potřeba a proto, je vložen do nižší úrovně a popsán až v tuto chvíli.

Uživatel si v systému nalezne konkrétní zakázku, ke které musí vyhledat dílce, z nichž se výrobek bude skládat. Pro tyto díly musí vytvořit plán výroby a vytisknout průvodní dokumentaci. V tomto kroku je dobré si povšimnout spouštěcí události typu message (Přijetí nové výrobní zakázky). Na tuto událost bude odkazováno v následujícím procesu kontroly kvality. Po vytištění průvodní dokumentace musí být rozhodnuto, jestli se jedná o výrobu hotových výrobků – pak jsou vytištěny výrobní štítky. V případě, že jde o výrobu polotovarů, tok pokračuje na podmínku, zdali se jedná o W zakázku. Výrobní zakázky jsou rozděleny podle počtu vyráběných kusů, v případě, že je jejich počet 20 a více, je výrobní zakázka označena písmenem W a musejí být vytištěny evidenční štítky. V jiném případě je pokračováno rovnou do zásobníku práce, ze kterého se nadále čerpá pro přiřazování úkolů jednotlivým pracovníkům.



Obrázek 151 Plánování výroby (zdroj: vlastní tvorba)

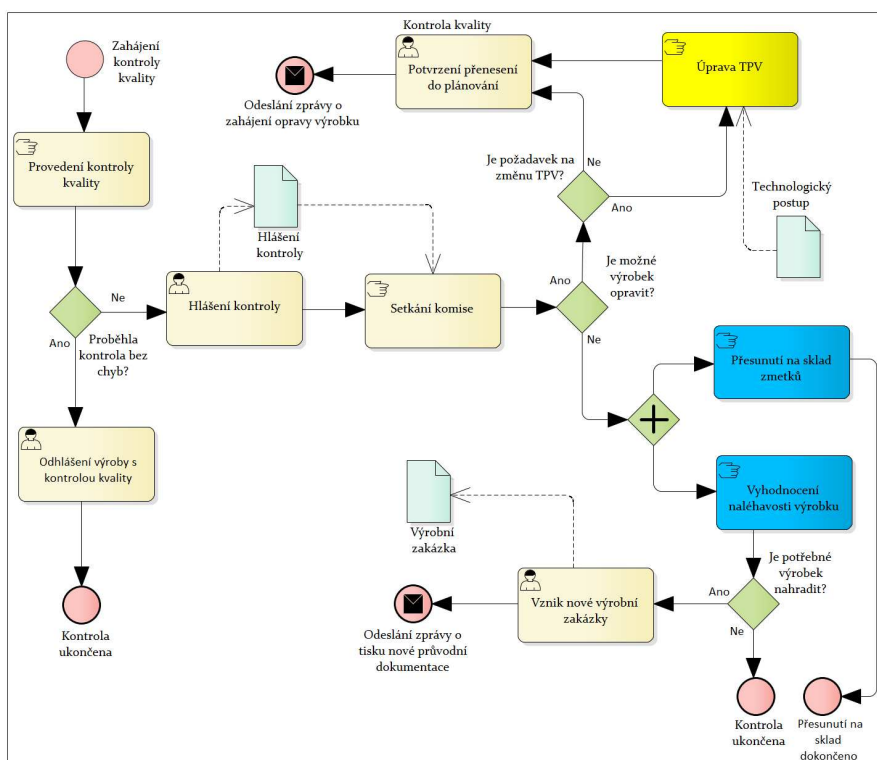
Kontrola kvality se zápisem do systému

Ve výrobě je možné vidět sub-proces provedení kontroly, který se provádí pro každý výrobek zvlášť. Proces je spuštěn ve chvíli, kdy zaměstnanec odevzdá výrobek jako hotový. Kontrolor provede kontrolu kvality, a jestliže kontrola proběhla v pořádku, odhlásí výrobek v systému s provedenou kontrolou. Tímto je proces kontroly ukončen a může pokračovat zpět ve výrobě. V případě, že kontrola výrobku neproběhla dle požadavků na výrobek, musí být zaspáno do systému hlášení o kontrole a je svolána komise, která posoudí další kroky průběhu výrobku. Pokud není výrobek možné opravit, jsou provedeny souběžně dvě manuální aktivity: Přenesení výrobku na sklad zmetků, kde je toto vlákno kontroly ukončeno a druhá aktivita Vyhodnocení naléhavosti výrobku. Zde je rozhodnuto, jestli je nutné nevyhovující výrobek znovu vyrobit nebo je možné se bez něho obejít. Toto rozhodnutí je možné provést, protože některé výrobky jsou plánovány takzvané podle ekonomických dávek. To znamená, že když uživatel plánuje výrobu, zadá takový počet výrobků, který bude nejvýhodnější vyrobit, vezmou-li se v potaz náklady na výrobu a uskladnění nevyužitých výrobků. Díky tomu může do výroby

vstoupit požadavek např. na 10 výrobků, z toho jsou 2 špatné, ale na zakázku je zapotřebí jenom 1 a tudíž není nutné vyrábět další výrobek znovu a toto vlákno je ukončeno.

V druhém případě uživatel vytvoří novou výrobní zakázku a ta je odeslána. Zde je možné si povšimnout koncové události typu message. V tomto kroku je vhodné si vzpomenout na plánování výroby a spouštěcí událost stejného typu, na kterou byl čtenář upozorněn. Zmíněná událost je spuštěna pouze událostí Odeslání zprávy o tisku průvodní dokumentace z Kontroly kvality. Příchodem této zprávy je znovu spuštěno plánování výroby, avšak nikoli na začátku procesu, ale od aktivity Tisk průvodní dokumentace, který spustí přijetí zprávy.

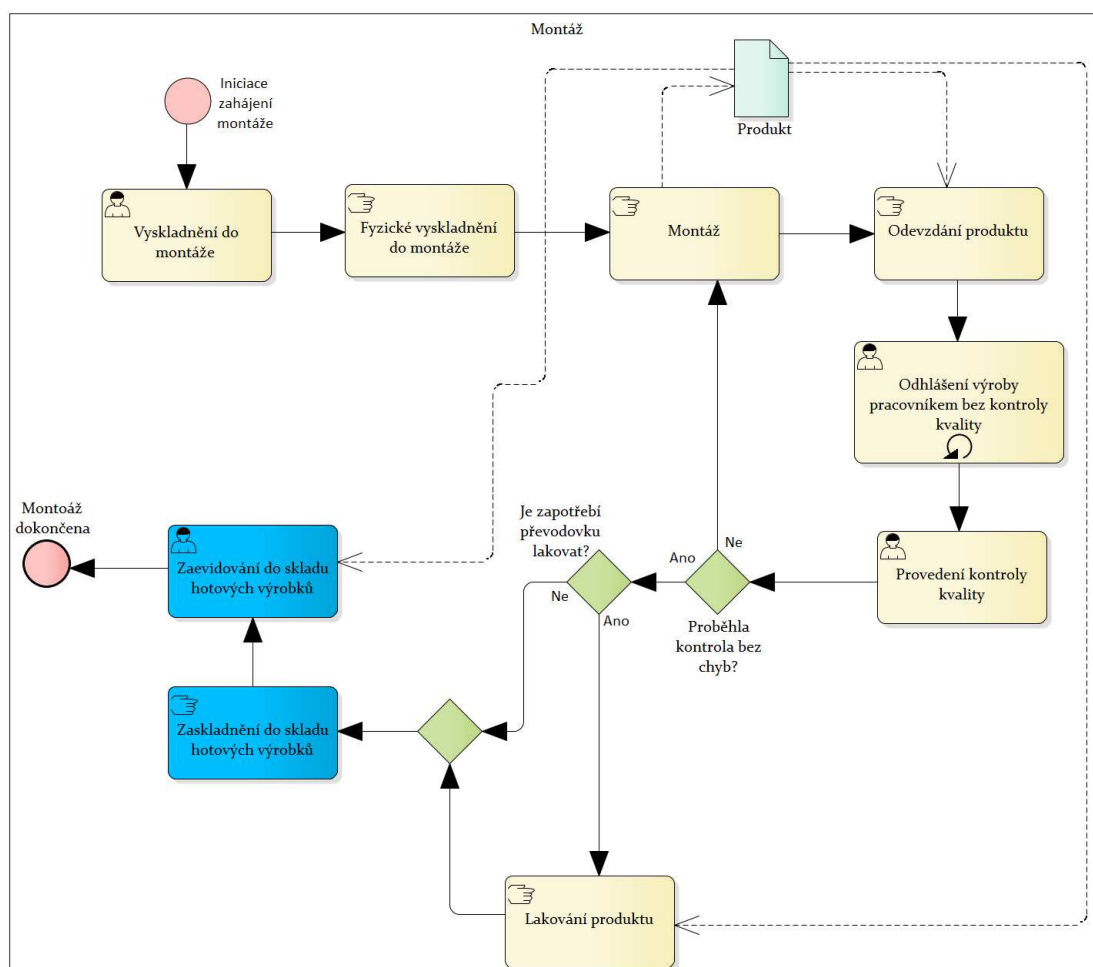
Je-li možné výrobek opravit, dojde k přezkoumání nutnosti provedení změny v TPV. Za předpokladu, že je zadán požadavek na úpravu TPV, je manuálně upraven technologický postup a předán do plánování. Vlákno procesu končí odesláním zprávy o zahájení opravy výrobku zpět do plánování výroby. Tato část je podobná vzniku nové zakázky. S tím rozdílem, že proces nezačíná tiskem průvodní dokumentace, ale upravený technologický postup je vložen přímo do zásobníku práce.



Obrázek 152 Kontrola kvality se zápisem do systému (zdroj: vlastní tvorba)

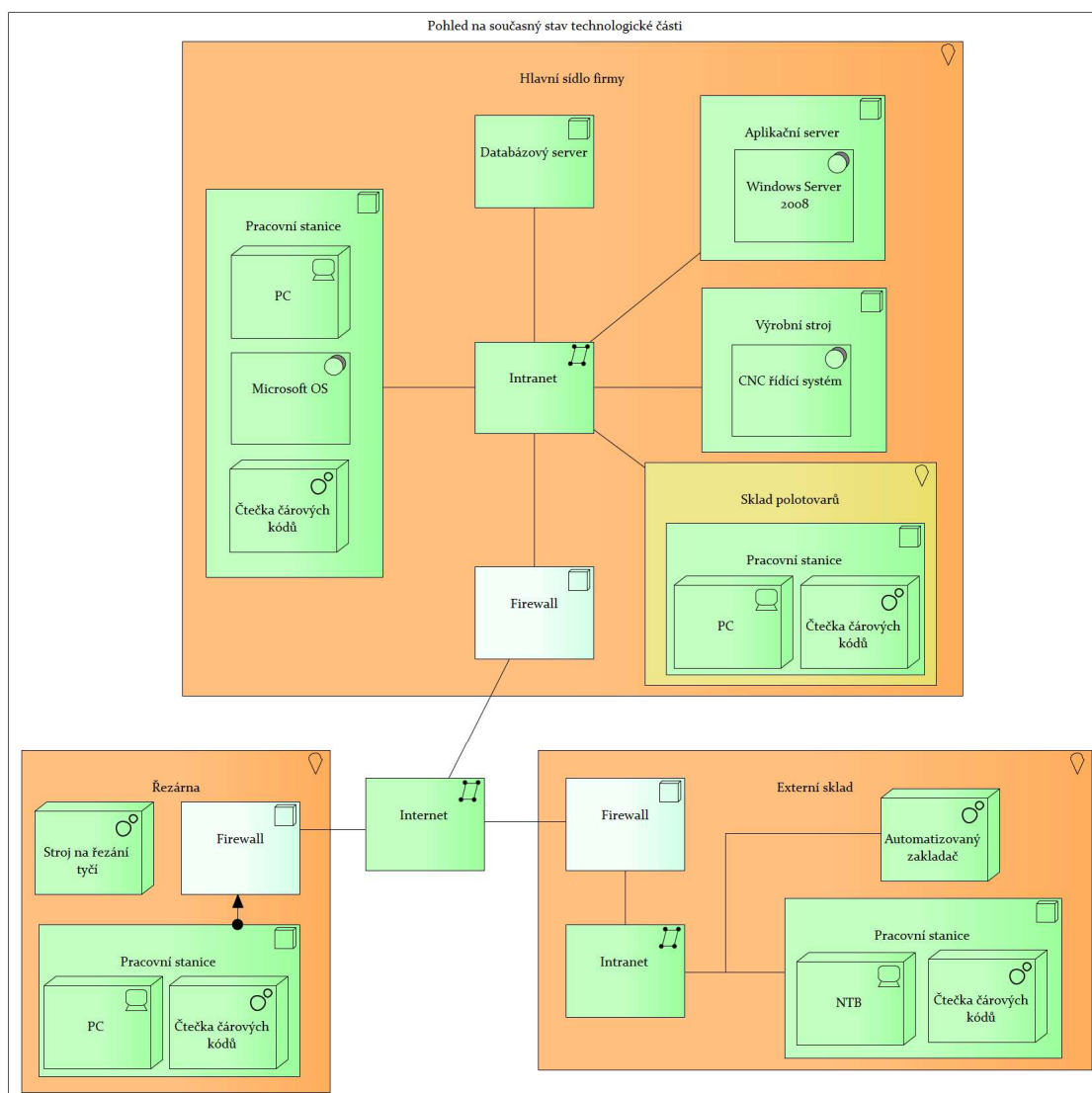
Montáž

Diagram znázorněný výše zobrazuje proces montáže. Montáž výrobku je zahájena poté, co jsou na skladně všechny části potřebné pro sestavení finálního výrobku. Proces zahajuje uživatel zadáním do systému materiál, který je potřeba vyskladnit na daný výrobek. Zadané výrobky jsou vyskladněny a dodány do montáže, kde probíhá finalizace výrobku do požadovaného stavu. Smontovaný výrobek je odevzdán na kontrolu kvality a do systému jsou zadány všechny úkony, které byly na výrobku provedeny. V dalším kroku je provedena kontrola kvality, jestli výrobek odpovídá požadavkům, podobně jako v části výroby. Není-li výrobek v pořádku, je předán zpět do montáže. U výrobku, který prošel kontrolou, je posouzeno, zda je dle požadavků nutné ho lakovat. V případě že ano, je nalakován a zaskladněn. V případě, že není potřeba výrobek lakovat, je rovnou převezen na sklad hotových výrobků a zaevidován do systému hotových výrobků.



Obrázek 153 Montáž (zdroj: vlastní tvorba)

Pohled na současný stav technologické části firmy



Obrázek 154 Pohled na současný stav technologické části (zdroj: vlastní tvorba)

Diagram technologické části popisuje současný stav technologické infrastruktury firmy, aby bylo jasné, kam všude bude nový systém zasahovat a jaký systém či hardware bude tento nový systém využívat. Z diagramu je patrné, že budovy výrobní firmy se nacházejí na třech různých lokacích a komunikace probíhá prostřednictvím internetu. To znamená, že systém bude muset být zabezpečen i pro komunikaci mimo intranet. Hlavní sídlo firmy obsahuje databázový a aplikační server, které jsou společně s pracovními stanicemi a výrobními stroji spojeny pomocí lokální sítě. Hlavní sídlo firmy komunikuje se sklady

prostřednictvím internetu, s výjimkou skladu polotovarů, který se nachází přímo v hlavním sídle. V druhé lokaci, která je nazvána Řezárna, se nachází pouze pracovní stanice se čtečkou čárových kódů a firewallem přistupující k serverům přes internet. Dále se zde nachází stroj na řezání, který pouze reprezentuje účel dané lokace, avšak se systémem není nijak propojen. Ve třetí lokaci Externí sklad je lokalizována pracovní stanice stejně jako ve všech ostatních lokacích. Navíc je tu jednoduchý automatický zakladač, zakládající zboží na základě čárových kódů.

4.6 Navržení cílového řešení

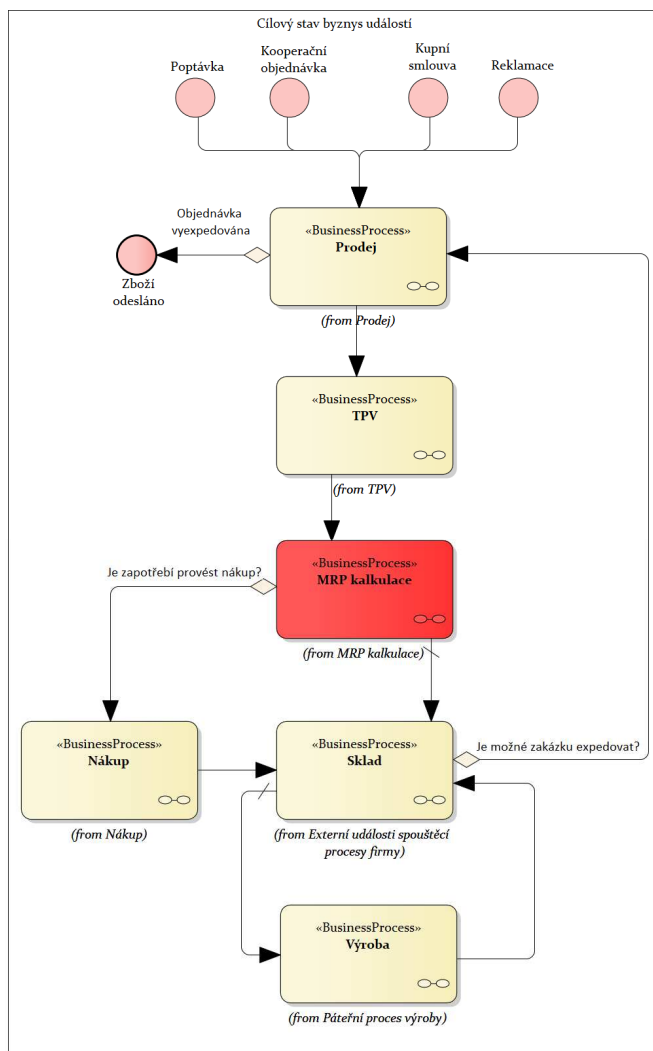
Tato kapitola se zabývá návrhem a úpravou firemních procesů a technologické části výrobní firmy. Jejím cílem je zdokonalení současného stavu na základě definovaných cílů. Z předchozích kapitol je patrné, že obecné procesy ERP systému a samotné procesy výrobní firmy se značně liší. Z velké části aktivit je však možné vyjít a tyto aktivity využít při realizaci nového systému. Při návrhu řešení nesmí dojít na transformaci a ke snaze o úpravu aktivit výrobního podniku, tak aby bylo, co nejjednodušší v něm nasadit systém, ale musí dojít k samotné úpravě systému. Tvorba systému bývá zákaznický orientována. To znamená, že současný systém je upravován, dle požadavků klienta tak, aby bylo dosaženo co možná největší spokojenosti a v dalších letech se obracel znovu na dotyčného dodavatele. Zákazník je však spokojen pouze tehdy, funguje-li systém dobře a pracovníci odvádějí kvalitní práci. A jelikož většina lidí špatně přijímá změny a každou novou věc bere jako zbytečnou, je důležité, aby se do značné míry neměnily lidské procesy, ale především docházelo ke změnám procesů, které provádí počítač.

Cílový stav podnikových procesů a technologické části je realizován promítnutím změn do modelů současného stavu firmy popsaného v předchozí kapitole. Jednotlivé byznys procesy již od sebe nebudou oddělovány barvami, tak jak bylo doporučováno, ale budou označeny pouze elementy nové, jež budou zvýrazněny červenou barvou. Modely, ze kterých je vycházeno, jsou stejné, jako v předchozí kapitole. Proto v této kapitole nebudou znovu detailně popisovány.

Dopady na byznys události

Oproti původnímu diagramu byznys procesů, zde přibyl jeden zcela nový proces MRP kalkulace, který umožní téměř plně nahradit aktivitu plánování výroby. Další

změnou je, že nákup není prováděn na základě TPV, ale je vytvořen na základě propočtů z MRP kalkulace. Tento proces bude celý vytvářet systém a tým, budou dodržena dvě základní pravidla. Pro uživatele v tuto chvíli nepřibude žádná práce a společnost se bude blížit k cíli automatizování svých procesů.

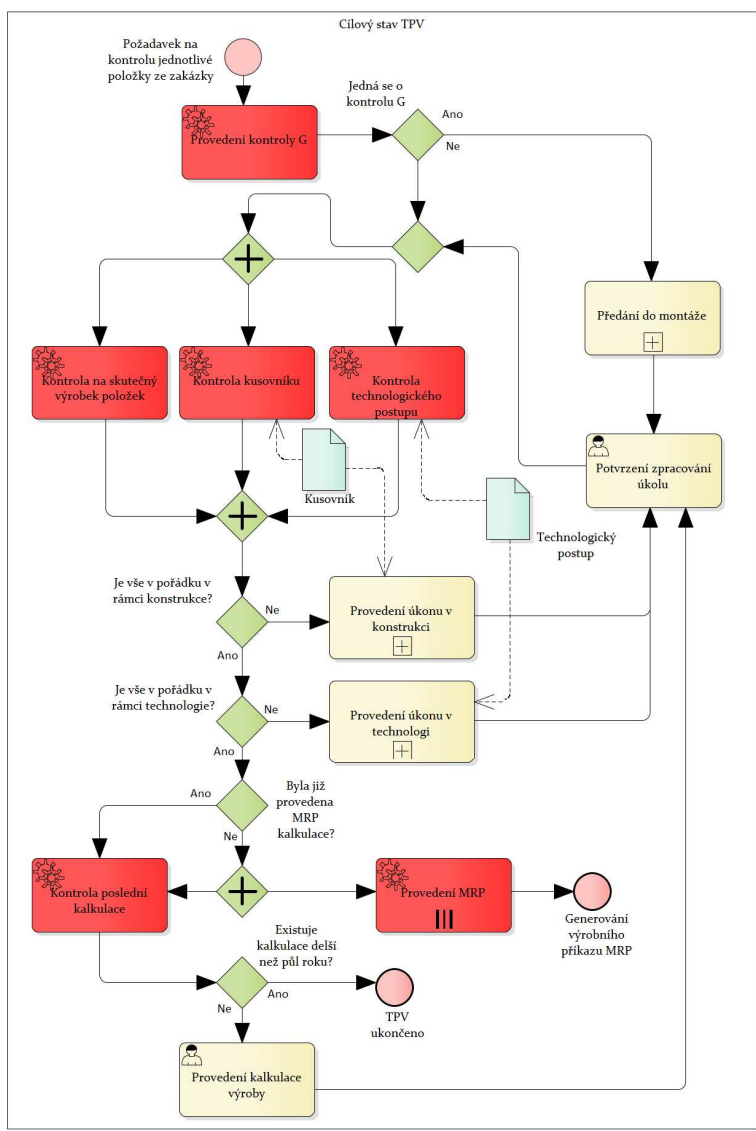


Obrázek 155 Cílový stav byznys událostí (zdroj: vlastní tvorba)

Dopady na technologickou přípravu výroby

Proces TPV vychází stejně jako model byznys procesů ze současného stavu firmy a snaží se udržet stejné procesy, které již ve firmě mají. Nicméně i tak v tomto procesu došlo ke spoustě změn, a to především k automatizaci některých aktivit. Hned po spuštění procesu je možné vidět, že aktivitu G neprovádí zaměstnanec, ale tato aktivita je nahrazena systémem. Rozhodne se, jestli se jedná o G zakázku a v případě že ne, tok pokračuje na tři paralelní aktivity, na které již nejsou potřeba

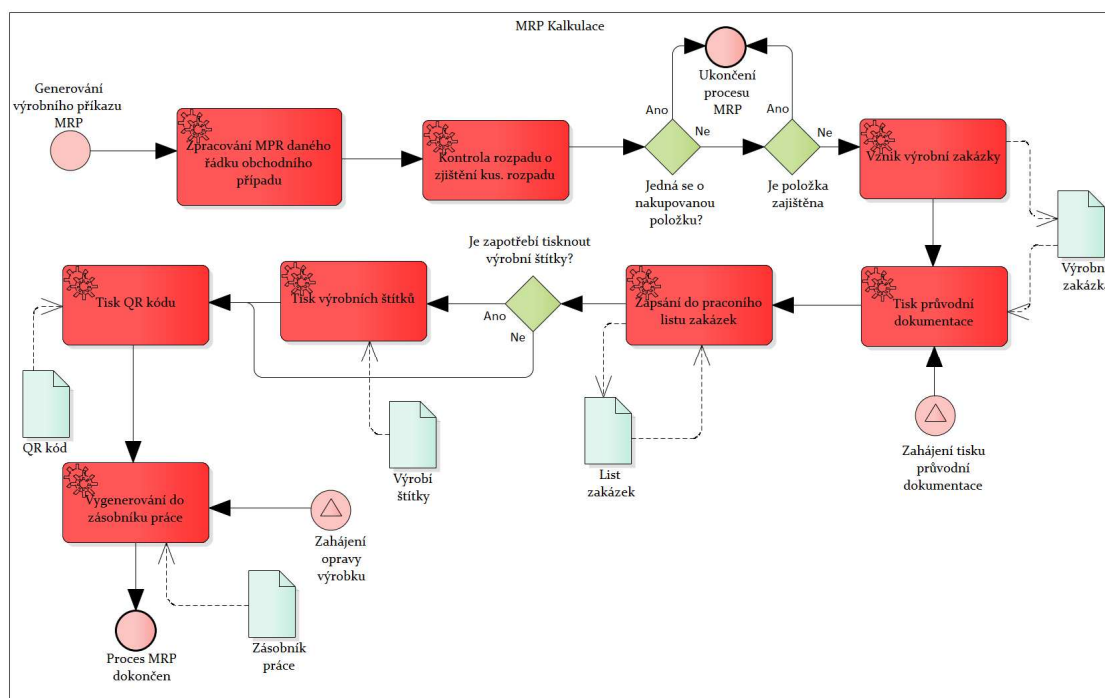
dva zaměstnanci, ale systém je provede sám. Také zde přibyla jedna aktivita, kterou je Kontrola na skutečný výrobek. V této aktivitě systém zkontroluje, jestli zakázka obsahuje všechny potřebné náležitosti jako je např. vyplněný název výrobku, zda je správně zadána skupina zboží nebo zdali je vyplněno číslo kusovníku. Tok pokračuje běžným způsobem přes několik podmínek, až dojde k nové podmínce stanovující, jestli již byla vyhotovena MRP kalkulace. V případě, že pro danou zakázku ještě kalkulace neproběhla, jsou spuštěny současně dvě nové aktivity. Samotná MRP kalkulace a kontrola poslední kalkulace nákladů na výrobu finálního produktu. V případě, že kalkulace na daný výrobek existuje a je mladší než půl roku, proces TPV je ukončen. V jiném případě musím kalkulant provést novou kalkulaci, potvrdit ji a proces se vrací zpět ke kontrolám.



Obrázek 156 Cílový stav TPV

MRP Kalkulace

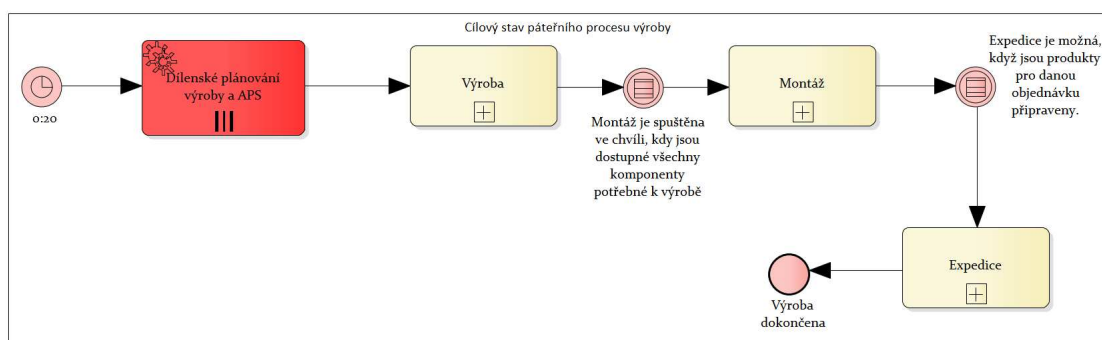
Jak již bylo zmíněno, tento proces je nový, avšak jeho značná podobnost může být viděna v procesu plánování výroby s tím rozdílem, že celý tento proces provádí pouze systém. Proces je spuštěn vygenerováním výrobního příkazu z procesu TPV a začne zpracovávat jednotlivé řádky obchodního příkazu. Poté se provede kontrola rozpadu o zjištění kusovníkového rozpadu, která vytvoří přehled, z jakých materiálů a polotovarů je daný dílec vyráběn. Zjistí se, jestli se jedná o nakupovanou položku, nebo daný výrobek je již skladem. V takovém případě by byla MRP kalkulační ukončena. V opačném případě je vytvořena výrobní zakázka, ke které je vytištěna průvodní dokumentace. V tomto procesu plánování výroby byl čtenář upozorněn na spouštěcí událost typu message. Zde je nahrazena událostí signál, kterou spouští jiná část systému. Po vytištění průvodní dokumentace je výrobní zakázka zapsána do pracovního listu zakázek. Po tomto kroku dle stejné podmínky je rozhodováno, zda mají být tištěny výrobní štítky, které generuje a tiskne systém. Evidenční štítky jsou již nahrazeny QR kódy a jsou tisknuty vždy pro každou zakázku. Proces MRP kalkulační končí vygenerováním zakázky do zásobníku práce.



Obrázek 157 MRP Kalkulace (zdroj: vlastní tvorba)

Cílový stav páteřního procesu výroby

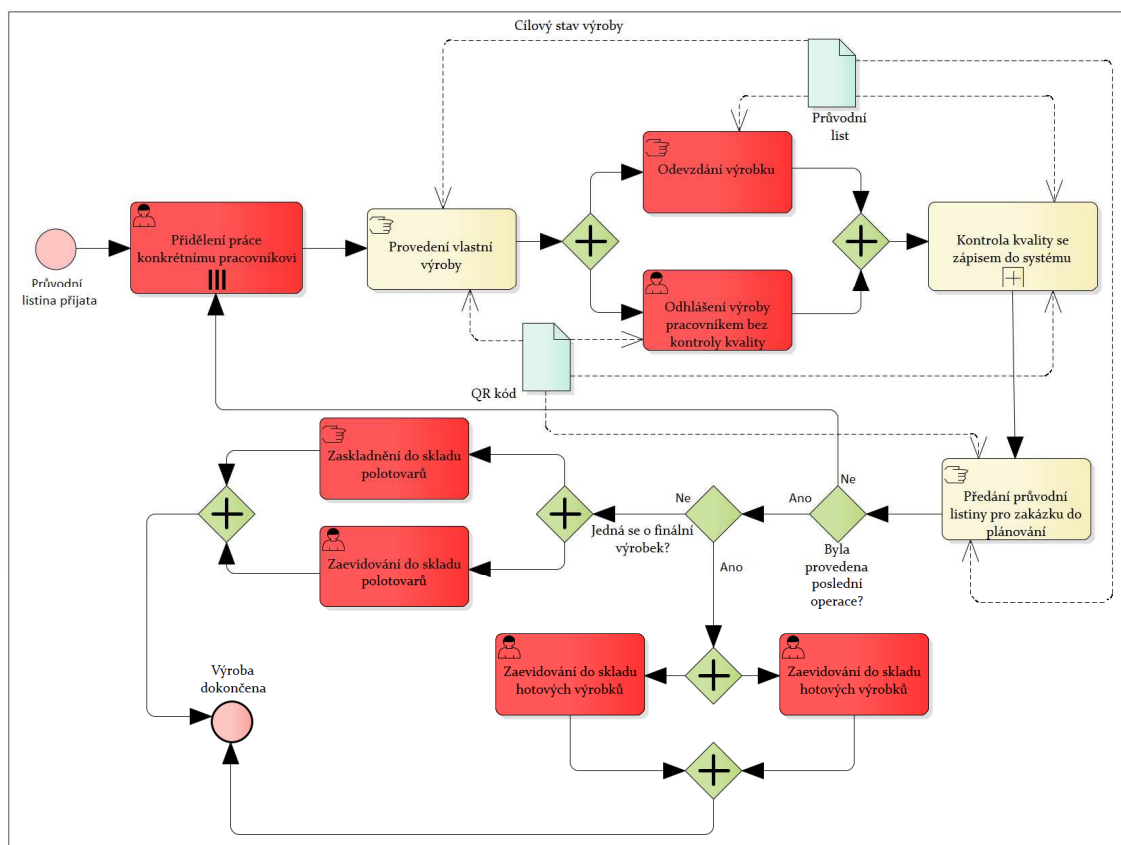
V páteřním procesu nastaly dvě změny. V první řadě je možné si všimnout spouštěcí události typu timer. Proces výroby je automaticky zahájen vždy 20 minut po půlnoci. Dále se zde objevuje aktivita Dílenské plánování, ve které systém naplňuje časy výroby k jednotlivým zakázkám a také přibližný počet zakázek, které by se za pracovní den měli zvládnout vyrobit. Pracovníci plánování tak nemusí plánovat celou výrobu, ale pouze přiřazují úkoly.



Obrázek 158 Cílový stav páteřního procesu výroby (zdroj: vlastní tvorba)

Cílový stav výroba

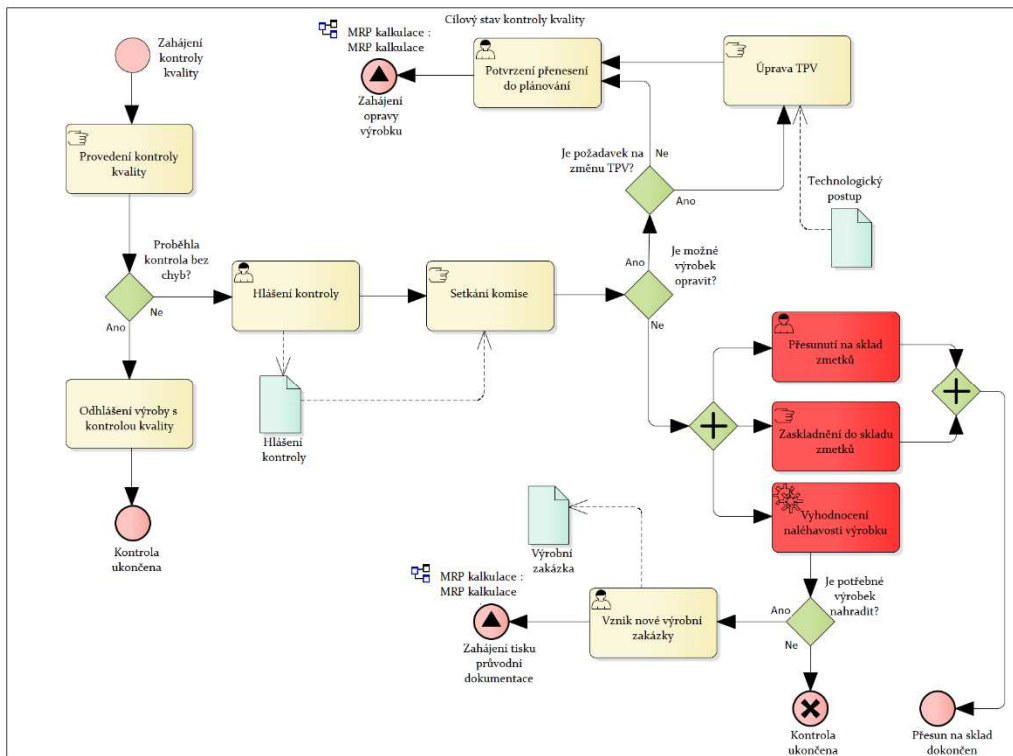
Výroba je zahájena přijetím průvodní listiny do výroby. Zde už jen pracovníci přidělí práci jednotlivým pracovníkům a ti mohou začít pracovat. Tato aktivita doposud nemohla být plně nahrazena systémem, protože při přiřazování práce je neustále zapotřebí lidský faktor a znalost pracovníků (např. je možné, že pracovník A vyrobí 3 výrobky za stejnou dobu, co pracovník B pouze 2). Další zásadní změnou je odevzdávání hotového výrobku. Pracovník už nebude muset nejprve jít fyzicky odevzdat výrobek a pak jej odhlásit v systému, ale tento krok bude možné vykonávat současně. To bude možné za použití přenosných QR čteček umístěných přímo v místě odevzdávání výrobku. Pracovník načte umístění, kam bude výrobek odkládat a současně i kód výrobku. To povede ke zjednodušení práce, ale také především k zabezpečení, že při montáži nebude použita jiná součástka, než do finálního výrobku má patřit.



Obrázek 159 Cílový stav výroby (zdroj: vlastní tvorba)

Cílový stav kontroly kvality

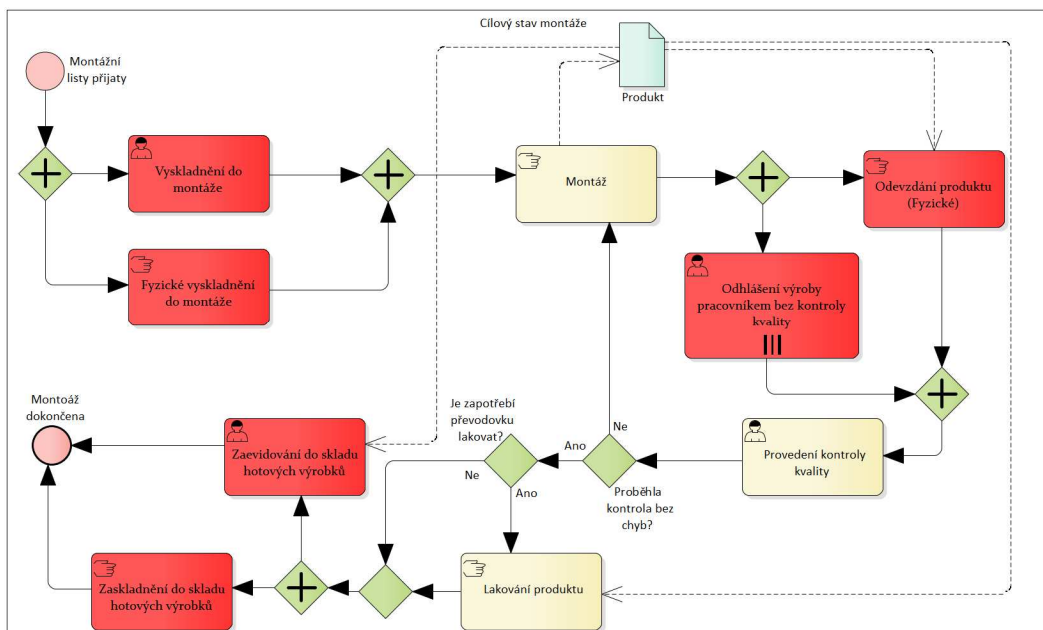
U kontroly kvality nedošlo k tak výrazným změnám jako u ostatních procesů. Změny se převážně týkají neopravitelných výrobků, takzvaných zmetků: jejich přesun na sklad zmetků je nově zaznamenáván i do systému. Pomocí QR čtečky lze tuto aktivitu vykonávat souběžně, stejně jako při odevzdávání jiných výrobků. Další změna nastane při vyhodnocování, zda je potřebné vadný výrobek vyrobit znovu. Tento úkon již nebude vytvářet pracovník, ale rozhodování bude prováděno automaticky systémem. Na pracovníky pouze je, aby vytvořil novou výrobní zakázku. Ke změně došlo i u dvou koncových událostí: z typu message na typ signal. Je to způsobeno tím, že pracovníci již neprovádějí plánování výroby, ale tento proces vykonává systém. Uživateli tak tedy stačí, aby pouze vytvořil novou zakázku nebo plán výroby a systém jej automaticky přesune do procesu MRP.



Obrázek 160 Cílový stav kontroly kvality (zdroj: vlastní tvorba)

Cílový stav montáže

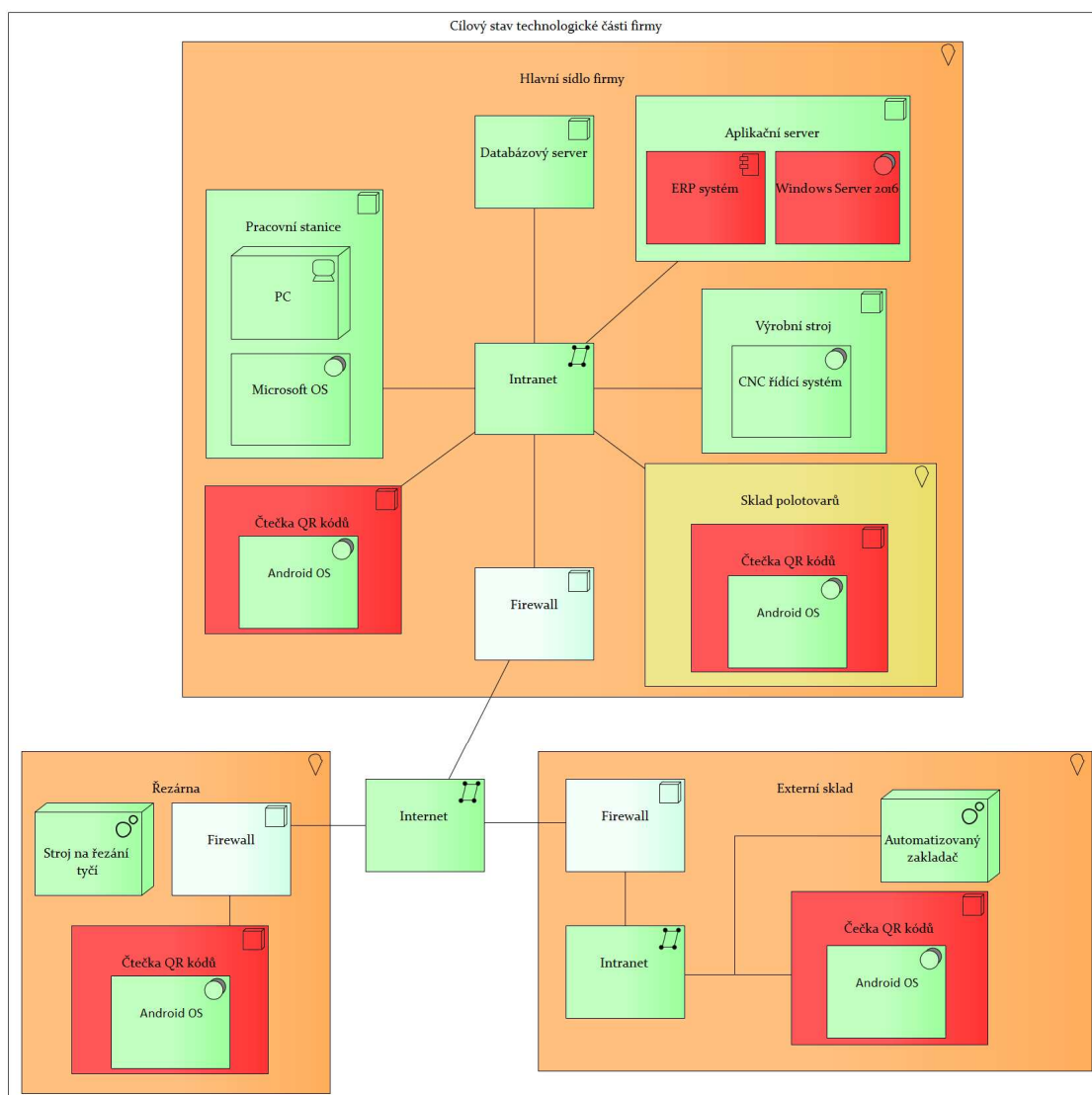
V montáži došlo pouze k zjednodušení vyskladňování a odevzdávání výrobků, a to zavedením QR kódů. Zbytek montáže z procesního hlediska zůstává beze změny.



Obrázek 161 Cílový stav montáže (zdroj: vlastní tvorba)

Cílový stav technologické části firmy

Aby mohlo dojít k samotné implementaci ERP systému, bude nutné nahradit stávající aplikační server novým výkonnějším serverem. Na tento server bude implementován ERP systém, ke kterému budou přistupovat jak stávající počítačové stanice, tak nově pořízené QR čtečky. Další provedenou změnou jsou již zmíněné QR čtečky, jež nahradí zastaralé počítačové stanice využívající pouze pro načtení čárových kódů.



Obrázek 162 Cílový stav technologické části (zdroj: vlastní tvorba)

4.6.1 Zhodnocení konečného stavu

Cílem je navržení a úprava podnikové architektury a podnikových procesů, tak aby mohl být zaveden nový systém a tím byla zlepšena dosavadní situace výrobní

společnosti. S tím je spojeno nejen zvýšení tržeb z prodeje, ale také automatizace a zjednodušení některých stávajících procesů, které mohly vést k chybovosti a prodávám ve výrobě. Propojení ERP systému společně s čtečkami QR kódů přinesou pozitivní efekt nejen pro výrobní společnost, ale také pro zaměstnance, kterým bude ulehčena část jejich práce. Vedení bude moci sledovat průběh různých operací a pohyb materiálu a výrobků, díky čemuž se zabrání použití špatných dílů alepší přehled pohybu ve firmě. Za pomoci odhadů a predikcí bude mnohem snazší plánování krátkodobých, ale i dlouhodobých cílů. S pomocí automatizace procesů může firma využít celého potenciálu pracovníků, kteří dosud byli zdržováni náročnými aktivitami a nemohli se plně věnovat své práci. Tomu také napomůže noční chod systému. Nastavení nových procesů spojených s implementací nového systému a změnou technologické infrastruktury s sebou však může přinášet i určitá rizika, která společnost před nasazením systému musí zvážit.

5 Shrnutí výsledků

V teoretické části autor vytvořil popis syntaxí BPMN a ArchiMate. Zpracování obou syntaxí bylo nejnáročnější pasáží práce. I přesto, že syntaxe BPMN je tu již dlouho a existuje mnoho článků, které se jí věnují, jsou mezi nimi neustále značné nesrovnalosti ohledně používání jednotlivých elementů.

Z toho důvodu se autor zaměřil především na oficiální dokumenty společnosti OMG, která danou syntaxi spravuje. Tento dokument, je však velmi složitý a nepřehledný.

Oproti tomu mnohem novější syntaxe ArchiMate má pro uživatele přívětivější dokumentaci. Tato dokumentace je však příliš obsáhlá a u popisu jednotlivých elementů lze zaznamenat mnohonásobné opakování stejných skutečností. I přes vyčerpávající rozsáhlost dokumentace dochází u některých z elementů k nejasnosti jejich interpretace a můžou být tak pro uživatele matoucí. Značnou nevýhodou je, že tato syntaxe dosud není tolik rozšířená a neexistují manuály, které by mohly neurčitosti upřesnit. Proto byly tyto syntaxe zjednodušeny a popsány tak, aby byly pro čtenáře přehledné, ale také dostatečně podrobné a přinášely ty správné relevantní informace.

Grafické modelování probíhalo ve třech krocích a celkově trvalo několik měsíců. Nejdříve byly na základě konzultací s analytiky a vývojáři vypracovány obecné procesy modulu výroby již existujícího ERP systému ve vývojové společnosti. Takto vytvořená šablona slouží vývojářům k usnadnění konstrukce již konkrétního systému, který je určen jako produkt pro zákazníka. V prvotních konzultacích s klientem slouží tento obecný model k seznámení s grafickým modelováním procesů a jeho výhodami.

V druhém kroku byl zmodelován aktuální stav procesů, který je důležitý pro vývojářskou společnost k pochopení filozofie zadavatelské firmy. Zároveň zanalyzuje velikost projektu a jeho náročnost. Pro tento krok je podstatné stanovení cílů, zákazník definuje požadavky na nový systém. Po vzájemné diskuzi vývojáři-zákazník se zhodnotí reálnost aplikace všech požadavků a ustanoví se závěry. V tomto kroku se zmodelovala technologická část klientské společnosti, která zanalyzuje, která zařízení budou do systému přistupovat a také jakým

způsobem. V tomto kroku nastaly problémy, které se týkaly komunikace a znalosti procesů ve firmě. Na konzultaci nebyl přítomen klíčový zaměstnanec, který měl popsat část procesu a jeho zástupce neměl o situaci tak dobrý přehled. To si však sám uvědomil až při samotném modelování. Textová dokumentace, jež by tento oříšek vyřešila, byla neaktuální. Tyto znalostní a komunikační nedostatky znemožnily zanalyzování současného stavu firmy a bylo nutné domluvit další konzultaci s více zasvěceným odborným pracovníkem. Po tomto problému sama zadavatelská firma připustila výhodnost grafického modelování oproti textové dokumentaci.

Třetím krokem bylo vytvoření návrhu, který se zabývá, jakým způsobem budou modifikovány firemní procesy zadavatelské společnosti po implementaci ERP systému. Navržení tohoto modelu pomůže nastítnit budoucí stav společnosti a také ujasnit nesrovnalosti vzniklé při stanovení a úpravě požadavků na systém. Díky tomuto modelu byly potenciální nedostatky podchyceny dříve, než začal samotný vývoj systému. Takto vytvořený grafický návrh přispěl k urychlení vývoje systému s pomocí jasně definovaných procesů, které vycházejí z obecných procesů ERP systému a zadavatelské společnosti. Grafické modelování také napomohlo předejít případným budoucím reklamacím, jelikož vše bylo postupně konzultováno a konečný stav byl odsouhlasen dříve, že byl finální produkt vytvořen.

6 Závěry a doporučení

Tato diplomová práce vznikla jako ucelený dokument popisující převedení podnikové architektury do grafické podoby. Vypracování této práce bylo náročné a autor často narážel na spoustu obtíží spojených s nedostatkem znalostí spojených s výrobní oblastí a také s nejednoznačností různých odborných publikací a článků. Navíc každý z autorů si na danou problematiku vytváří svůj vlastní pohled, který vnáší i do své práce. Vytvořit si tak komplexní pohled na výrobní oblast pouze z literatury je nelehkým a komplikovaným úkolem. Jako nejvhodnější se proto časem ukázalo nevycházet pouze z teoretických hledisek, ale snažit se problematiku zrealizovat a popsat praktickými zkušenostmi. Mnohdy až ty pomohou čtenáři objasnit skutečné principy fungování zkoumané oblasti.

Pro vytváření modelů byl použit *case* nástroj Enterprise Architect, který je jediným sofistikovaným instrumentem podporujícím více modelovacích jazyků potřebných k této práci. Zároveň podporuje širokou paletu funkcí, které jsou vhodné pro řízení projektů. Prostřednictvím této práce její autor nabízí čtenářům další pohled na grafické modelování a přidává postupy využitelné při mapování podnikové architektury.

Teoretická část je zaměřena na seznámení se s výrobními metodami a syntaxemi BPMN a ArchiMate, což slouží k pochopení samotného modelování a porozumění praktické části. V té byly vytvořeny grafické modely. Tento postup byl pilotním testem vývojářské společnosti, který měl usnadnit návrh a implementaci ERP systému a přinesl mnohá zjištění. Prvotní zpracování postupů a vytvoření metodik bylo zpočátku časově i finančně velmi náročné. Po této fázi se však začaly objevovat pozitiva. Došlo k celkovému urychlení zakázkového vývoje ERP systému. Díky průběžným konzultacím a návrhům řešení systému se předešlo nedorozuměním a reklamacím. Obě společnosti vyhodnotily společný projekt využívající grafické modelování jako přínos.

Pro dokonalejší zmapování podnikové architektury by se dále dalo využít dalších diagramů v rámci jazyku ArchiMate: například diagram organizační struktury, podnikových norem, ale především síťové a datové infrastruktury, které v současné době mohou sloužit pro analýzu bezpečnosti dat v rámci GDPR

(General Data Protection Regulation – nařízení na ochranu osobních údajů, které začne v celé EU platit od 25. května 2018).

Procesy modelované pomocí jazyka BPMN by bylo možné rozšířit o simulace podnikových procesů, které mohou ukázat, jak je lze optimalizovat tak, aby byly pro firmu co možná nejefektivnější. V diplomové práci představené jazyky mohou být dále obohaceny o jazyk UML, který by celému modelování přidal na komplexnosti a umožnil tak zachytit více úrovní podnikové architektury.

7 Seznam použité literatury

- [1] KEŘKOVSKÝ, Miloslav a Ondřej VALSA. *Moderní přístupy k řízení výroby*. 3., dopl. vyd. V Praze: C.H. Beck, 2012. C.H. Beck pro praxi. ISBN 978-80-7179-319-9.
- [2] GRASSEOVÁ, Monika, Radek DUBEC a Roman HORÁK. *Procesní řízení ve veřejném sektoru: teoretická východiska a praktické příklady*. Brno: Computer Press, 2008. ISBN 978-80-251-1987-7. Název. Název odpovědné korporace, instituce. Roky vydání (od-do), ročník (od-do). Místo vydání: Název nakladatelství. Standardní číslo (ISSN).
- [3] CARDA, Antonín a Renata KUNSTOVÁ. *Workflow: řízení firemních procesů*. Praha: Grada, 2001. Management v informační společnosti. ISBN isbn80-247-0200-2.
- [4] FIŠER, Roman. *Procesní řízení pro manažery: jak zařídit, aby lidé věděli, chtěli, uměli i mohli*. Praha: Grada, 2014. Manažer. ISBN 978-80-247-5038-5.
- [5] KUCHARČÍKOVÁ, Alžbeta. *Efektivní výroba: využijte výrobní faktory a připravte se na změny na trzích*. Brno: Computer Press, 2011. ISBN 978-80-251-2524-3.
- [6] JUROVÁ, Marie. *Výrobní a logistické procesy v podnikání*. Praha: Grada Publishing, 2016. Expert (Grada). ISBN 978-80-247-5717-9.
- [7] KAVAN, Michal. *Výrobní a provozní management*. Praha: Grada, 2002. Expert (Grada). ISBN 80-247-0199-5.
- [8] JUROVÁ, Marie. *Řízení výroby*. Brno: Akademické nakladatelství CERM, 2011. ISBN 978-80-214-4370-9.
- [9] TOMEK, Gustav a Věra VÁVROVÁ. *Řízení výroby a nákupu*. Praha: Grada, 2007. Expert (Grada). ISBN 978-80-247-1479-0.
- [10] HORVÁTH, Gejza. *Logistika ve výrobním podniku*. V Plzni: Západočeská univerzita, 2007. ISBN 978-80-7043-634-9.
- [11] ŘEPA, Václav. *Podnikové procesy: procesní řízení a modelování*. 2., aktualiz. a rozš. vyd. Praha: Grada, 2007. Management v informační společnosti. ISBN 978-80-247-2252-8.

Odborné články

- [12]DOUGLAS K BARRY. Business Process Modeling Initiative (BPMI.org). *Service Architecture* [online]. 2005 [cit. 2018-02-06]. Dostupné z: https://www.service-architecture.com/articles/web-services/business_process_modeling_initiative_bpmi.org.html
- [13]TRISOTECH. *BPMN Introduction and History* [online]. 2015 [cit. 2018-02-06]. Dostupné z: <http://www.trisotech.com/articles/bpmn-introduction-history>
- [14]BPMI.ORG. *Business Process Modeling Notation* [online]. 2002 [cit. 2018-02-06]. Dostupné z: <http://xml.coverpages.org/BPMN-NWG200211.pdf>
- [15]WHITE A., Stephan. *BPMN: Past, Present, and Future* [online]. 2012 [cit. 2018-02-06]. Dostupné z: <http://bpm2012.ut.ee/wp-content/uploads/2012/03/keynote-white.pdf>
- [16]OMG. *Business Process Model and Notation (BPMN), V1.1* [online]. 2008 [cit. 2018-02-06]. Dostupné z: <http://www.omg.org/spec/BPMN/1.1>
- [17]OMG. *Business Process Model and Notation (BPMN), V1.2* [online]. 2009 [cit. 2018-02-06]. Dostupné z: <http://www.omg.org/spec/BPMN/1.2>
- [18]OMG. *Business Process Model and Notation (BPMN), V2.0.2* [online]. 2014 [cit. 2018-02-06]. Dostupné z: <http://www.omg.org/spec/BPMN/>
- [19]DOMBROSKI, Stephan a Lukáš DOLNÍČEK. Štíhlé principy a procesně orientovaná výroba. *SystemOnline* [online]. 2013 [cit. 2018-02-16]. Dostupné z: <https://www.systemonline.cz/rizeni-vyroby/stihle-principy-a-procesne-orientovana-vyroba.htm>
- [20]SPARX SYSTEMS. *Enterprise Architect 13: Reviewer's Guide* [online]. 2016 [cit. 2018-02-21]. Dostupné z: https://www.sparxsystems.eu/fileadmin/user_upload/pdfs/Enterprise_Architect_13_executive_summaryx EN.pdf

- [21]Informace. In: ManagementMania.com [online]. Wilmington (DE) 2011-2018, 14.12.2017 [cit. 21.02.2018]. Dostupné z: <https://managementmania.com/cs/informace>
- [22]JIT (Just-in-time). In: ManagementMania.com [online]. Wilmington (DE) 2011-2018, 23.06.2016 [cit. 21.02.2018]. Dostupné z: <https://managementmania.com/cs/just-in-time>
- [23]ERP Software. In: ManagementMania.com [online], 10.8.2016 [cit. 21.02.2018]. Dostupné z: <https://managementmania.com/cs/erp-system>
- [24]Řízení výroby (Production Management). In: ManagementMania.com [online]. Wilmington (DE) 2011-2018, 25.01.2016 [cit. 21.02.2018]. Dostupné z: <https://managementmania.com/cs/rizeni-vyroby>
- [25]THE OPEN GROUP. *Welcome to the ArchiMate® 3.0.1 Specification, an Open Group Standard* [online]. c2012-2017 [cit. 2018-02-09]. Dostupné z: <http://pubs.opengroup.org/architecture/archimate3-doc/toc.html>
- [26]THE OPEN GROUP. [online]. c2012-2013 [cit. 2018-02-09]. Dostupné z: <http://pubs.opengroup.org/architecture/archimate2-doc/toc.html>
- [27]THE OPEN GROUP. *THE Open Group* [online]. [cit. 2018-02-09]. Dostupné z: <http://www.opengroup.org/>
- [28]TAYLLORCOX. *ARCHIMATE* [online]. [cit. 2018-02-09]. Dostupné z: <http://www.tcox.cz/slovník/archimate/#>
- [29]THE OPEN GROUP. *The ArchiMate Enterprise Architecture Modeling Language* [online]. c1995-2018 [cit. 2018-02-09]. Dostupné z: <http://www.opengroup.org/subjectareas/enterprise/archimate-overview>
- [30]OMG Unified Modeling Language TM (OMG UML): Specification, Version 2.5 (formal/2015-03-01), Object Management Group, March 2015. Dostupné z: <http://www.omg.org/spec/UML/About-UML/>

Webové stránky

- [31]ISO/IEC JTC 1. *International Organization for Standardization: Information technology -- Object Management Group Business Process Model and Notation* [online]. 2013 [cit. 2018-02-19].
- [32]Bizagi: *Best practices in modeling* [online]. c2002-2018 [cit. 2018-03-07].
Dostupné z: http://help.bizagi.com/process-modeler/en/index.html?best_practices_in_modeling.htm
- [33]OMG. *Object Management Group* [online]. c2018 [cit. 2018-03-12].
Dostupné z: <https://www.omg.org/>
- [34]Archi. *The Free ArchiMate Modelling Tool* [online]. c2013-2018 [cit. 2018-03-12]. Dostupní z: <https://www.archimatetool.com/>
- [35]THE OPEN GROUP [online]. [cit. 2018-03-12]. Dostupné z: <http://www.opengroup.org/>
- [36]SPARX SYSTEMS. *Enterprise Architect* [online]. c2000-2018 [cit. 2018-03-12]. Dostupné z: <http://www.sparxsystems.com/products/ea/index.html>

Seznam obrázků

Obrázek 1 Druhy událostí (zdroj: [17], vlastní úprava).....	17
Obrázek 2 Non-Interrupting a Interrupting Event (zdroj: [17], vlastní úprava)	17
Obrázek 3 Zpráva (zdroj: [17], vlastní úprava).....	18
Obrázek 4 Časovač (zdroj: [17], vlastní úprava)	18
Obrázek 5 Chyba (zdroj: [17], vlastní úprava)	19
Obrázek 6 Eskalace (zdroj: [17], vlastní úprava)	19
Obrázek 7 Ukončení (zdroj: [17], vlastní úprava).....	20
Obrázek 8 Navrácení (zdroj: [17], vlastní úprava)	20
Obrázek 9 Podmínka (zdroj: [17], vlastní úprava)	20
Obrázek 10 Odkaz (zdroj: [17], vlastní úprava)	21
Obrázek 11 Signál (zdroj: [17], vlastní úprava).....	21
Obrázek 12 Ukončení (zdroj: [17], vlastní úprava)	21
Obrázek 13 Vícenásobný spouštěč (zdroj: [17], vlastní úprava)	21
Obrázek 14 Paralelní Spouštěč (zdroj: [17], vlastní úprava).....	22
Obrázek 15 Business Process (zdroj: [17], vlastní úprava)	22
Obrázek 16 Sub-proces (zdroj: [17], vlastní úprava)	23
Obrázek 17 Úkol (zdroj: [17], vlastní úprava).....	23
Obrázek 18 Multi-Instance (zdroj: [17], vlastní úprava).....	24
Obrázek 19 Loop (zdroj: [17], vlastní úprava)	24
Obrázek 20 Compensation Task (zdroj: [17], vlastní úprava)	24
Obrázek 21 Service Task (zdroj: [17], vlastní úprava).....	25
Obrázek 22 Send Task (zdroj: [17], vlastní úprava)	25
Obrázek 23 Receive Task (zdroj: [17], vlastní úprava).....	26
Obrázek 24 User Task (zdroj: [17], vlastní úprava)	26
Obrázek 25 Manual Task (zdroj: [17], vlastní úprava)	26
Obrázek 26 Rule Task (zdroj: [17], vlastní úprava)	26
Obrázek 27 Script Task (zdroj: [17], vlastní úprava)	27
Obrázek 28 Exclusive Gateway (zdroj: [17], vlastní úprava)	27
Obrázek 29 Inclusive Gateway (zdroj: [17], vlastní úprava)	28
Obrázek 30 Parallel Gateway (zdroj: [17], vlastní úprava)	28

Obrázek 31 Event-Base Gateway (zdroj: [17], vlastní úprava).....	29
Obrázek 32 Parallel Event-Base Gateway (zdroj: [17], vlastní úprava)	29
Obrázek 33 Complex Gateway (zdroj: [17], vlastní úprava).....	30
Obrázek 34 Sequence Flow (zdroj: [17], vlastní úprava).....	30
Obrázek 35 Conditional Sequence Flow (zdroj: [17], vlastní úprava)	30
Obrázek 36 Default Sequence Flow (zdroj: [17], vlastní úprava)	31
Obrázek 37 Message Flow (zdroj: [17], vlastní úprava).....	31
Obrázek 38 Association (zdroj: [17], vlastní úprava).....	31
Obrázek 39 Data Association (zdroj: [17], vlastní úprava)	31
Obrázek 40 Pool (zdroj: [17], vlastní úprava)	32
Obrázek 41 Lane (zdroj: [17], vlastní úprava)	32
Obrázek 42 Group (zdroj: [17], vlastní úprava)	33
Obrázek 43 Text Annotation (zdroj: [17], vlastní úprava)	33
Obrázek 44 Datový objekt (zdroj: [17], vlastní úprava).....	34
Obrázek 45 Data Store (zdroj: [17], vlastní úprava).....	34
Obrázek 46 Vztah mezi jazykem ArchiMate a metodikou TOGAF (zdroj: [24], vlastní úprava).....	35
Obrázek 47 Metamodel motivačních elementů (zdroj: [24], vlastní úprava)	36
Obrázek 48 Zainteresané strany (zdroj: [24], vlastní úprava).....	37
Obrázek 49 Iniciátor (zdroj: [24], vlastní úprava).....	37
Obrázek 50 Posouzení (zdroj: [24], vlastní úprava)	38
Obrázek 51 Cíl (zdroj: [24], vlastní úprava).....	38
Obrázek 52 Výsledek (zdroj: [24], vlastní úprava).....	38
Obrázek 53 Princip (zdroj: [24], vlastní úprava)	39
Obrázek 54 Požadavek (zdroj: [24], vlastní úprava)	39
Obrázek 55 Omezení (zdroj: [24], vlastní úprava)	40
Obrázek 56 Význam (zdroj: [24], vlastní úprava).....	40
Obrázek 57 Hodnota (zdroj: [24], vlastní úprava).....	40
Obrázek 58 Metamodel byznys vrstvy (zdroj: [24], vlastní úprava)	41
Obrázek 59 Byznys elementy aktivní struktury (zdroj: [24], vlastní úprava)	42
Obrázek 60 Byznys aktér (zdroj: [24], vlastní úprava).....	42
Obrázek 61 Byznys role (zdroj: [24], vlastní úprava)	43

Obrázek 62 Byznys součinnost (zdroj: [24], vlastní úprava)	43
Obrázek 63 Byznys interakce (zdroj: [24], vlastní úprava)	44
Obrázek 64 Byznys elementy chování (zdroj: [24], vlastní úprava).....	45
Obrázek 65 Byznys proces (zdroj: [24], vlastní úprava)	45
Obrázek 66 Byznys funkce (zdroj: [24], vlastní úprava)	46
Obrázek 67 Byznys interakce (zdroj: [24], vlastní úprava)	46
Obrázek 68 Byznys služba (zdroj: [24], vlastní úprava)	47
Obrázek 69 Byznys událost (zdroj: [24], vlastní úprava).....	47
Obrázek 70 Byznys objekt (zdroj: [24], vlastní úprava).....	48
Obrázek 71 Byznys kontrakt (zdroj: [24], vlastní úprava).....	48
Obrázek 72 Reprezentace (zdroj: [24], vlastní úprava)	49
Obrázek 73 Produkt (zdroj: [24], vlastní úprava).....	49
Obrázek 74 Metamodel Aplikační vrstvy (zdroj: [24], vlastní úprava)	50
Obrázek 75 Aplikační komponenta (zdroj: [24], vlastní úprava)	51
Obrázek 76 Aplikační součinnost (zdroj: [24], vlastní úprava).....	51
Obrázek 77 Aplikační rozhraní (zdroj: [24], vlastní úprava)	52
Obrázek 78 Aplikační funkce (zdroj: [24], vlastní úprava)	52
Obrázek 79 Aplikační interakce (zdroj: [24], vlastní úprava).....	53
Obrázek 80 Aplikační proces (zdroj: [24], vlastní úprava)	53
Obrázek 81 Aplikační událost (zdroj: [24], vlastní úprava).....	54
Obrázek 82 Aplikační služba (zdroj: [24], vlastní úprava).....	54
Obrázek 83 Datový objekt (zdroj: [24], vlastní úprava).....	55
Obrázek 84 Metamodel Aplikační vrstvy (zdroj: [24], vlastní úprava)	55
Obrázek 85 Uzel (zdroj: [24], vlastní úprava)	56
Obrázek 86 Zařízení (zdroj: [24], vlastní úprava).....	56
Obrázek 87 Systémový software (zdroj: [24], vlastní úprava)	57
Obrázek 88 Technologická součinnost (zdroj: [24], vlastní úprava).....	57
Obrázek 89 Technologické rozhraní (zdroj: [24], vlastní úprava)	58
Obrázek 90 Cesta (zdroj: [24], vlastní úprava)	58
Obrázek 91 Komunikační síť (zdroj: [24], vlastní úprava).....	58
Obrázek 92 Technologická funkce (zdroj: [24], vlastní úprava).....	59
Obrázek 93 Technologický proces (zdroj: [24], vlastní úprava).....	59

Obrázek 94 Technologická interakce (zdroj: [24], vlastní úprava).....	60
Obrázek 95 Technologická událost (zdroj: [24], vlastní úprava).....	60
Obrázek 96 Technologická služba (zdroj: [24], vlastní úprava).....	61
Obrázek 97 Technologický objekt (zdroj: [24], vlastní úprava).....	61
Obrázek 98 Artefakt (zdroj: [24], vlastní úprava)	62
Obrázek 99 Metamodel Fyzických prvků (zdroj: [24], vlastní úprava).....	62
Obrázek 100 Vybavení (zdroj: [24], vlastní úprava).....	63
Obrázek 101 Prostředí (zdroj: [24], vlastní úprava)	63
Obrázek 102 Distribuční síť (zdroj: [24], vlastní úprava)	64
Obrázek 103 Materiál (zdroj: [24], vlastní úprava).....	64
Obrázek 104 Lokace (zdroj: [24], vlastní úprava)	65
Obrázek 105 Přehled vztahů (zdroj: [24], vlastní úprava).....	65
Obrázek 106 Kompozice (zdroj: [24], vlastní úprava)	67
Obrázek 107 Příklad vazby kompozice (zdroj: vlastní tvorba)	67
Obrázek 109 Agregace (zdroj: [24], vlastní úprava).....	68
Obrázek 108 Příklad vazby agregace (zdroj: vlastní tvorba)	68
Obrázek 110 Přiřazení (zdroj: [24], vlastní úprava).....	68
Obrázek 111 Příklad vazby přiřazení (zdroj: vlastní tvorba).....	69
Obrázek 112 Realizace (zdroj: [24], vlastní úprava)	69
Obrázek 113 Příklad vazby realizace (zdroj: vlastní tvorba)	69
Obrázek 114 Sloužící (zdroj: [24], vlastní úprava).....	70
Obrázek 115 Příklad sloužící vazby (zdroj: vlastní tvorba).....	70
Obrázek 116 Přístup (zdroj: [24], vlastní úprava).....	71
Obrázek 117 Příklad vazby přístup (zdroj: vlastní tvorba)	71
Obrázek 118 Ovlivnění (zdroj: [24], vlastní úprava)	71
Obrázek 119 Příklad vazby ovlivnění (zdroj: vlastní tvorba)	72
Obrázek 120 Spuštění (zdroj: [24], vlastní úprava).....	72
Obrázek 121 Příklad vazby spuštění (zdroj: vlastní tvorba)	72
Obrázek 122 Tok (zdroj: [24], vlastní úprava).....	73
Obrázek 123 Příklad vazby tok (zdroj: [24], vlastní úprava)	73
Obrázek 124 Specializace (zdroj: [24], vlastní úprava).....	73
Obrázek 125 Příklad vazby specializace (zdroj: vlastní tvorba).....	73

Obrázek 126 Asociace (zdroj: [24], vlastní úprava).....	74
Obrázek 127 Příklad vazby asociace (zdroj: vlastní tvorba)	74
Obrázek 128 Spojka (zdroj: [24], vlastní úprava).....	75
Obrázek 129 Seskupení zdroj: vlastní tvorba)	75
Obrázek 130 Project Browser (zdroj: vlastní tvorba)	77
Obrázek 131 Rozpad při tvorbě procesů (zdroj: vlastní tvorba)	82
Obrázek 132 MRP kalkulace dle Happy path (zdroj: vlastní tvorba)	84
Obrázek 133 Rozšíření MRP kalkulace (zdroj: vlastní tvorba).....	84
Obrázek 134 MRP kalkulace rozšíření o datové objekty (zdroj: vlastní tvorba)	85
Obrázek 135 Organizační struktura obchodního modelu (zdroj: vlastní tvorba)....	87
Obrázek 136 Hlavní proces modulu výroba (zdroj: vlastní tvorba)	88
Obrázek 137 Konfigurace systému (zdroj: vlastní tvorba)	89
Obrázek 138 Plánování výroby (zdroj: vlastní tvorba)	90
Obrázek 139 MRP kalkulace (zdroj: vlastní tvorba)	91
Obrázek 140 Zpracování požadavku na výrobu (zdroj: vlastní tvorba)	92
Obrázek 141 Řízení výroby (zdroj: vlastní tvorba).....	93
Obrázek 142 Operativní řízení výroby (zdroj: vlastní tvorba)	94
Obrázek 143 Odvádění výkonů výroby (zdroj: vlastní tvorba)	95
Obrázek 144 Vyhodnocení výroby (zdroj: vlastní tvorba).....	96
Obrázek 145 Moduly a služby ERP systému (zdroj: vlastní tvorba).....	97
Obrázek 146 Cíle (zdroj: vlastní tvorba)	99
Obrázek 147 Události a byznys procesy výrobní firmy (zdroj: vlastní tvorba)	100
Obrázek 148 Technologická příprava výroby (zdroj: vlastní tvorba).....	101
Obrázek 149 Páteční proces výroby (zdroj: vlastní tvorba)	103
Obrázek 150 Výroba (zdroj: vlastní tvorba)	104
Obrázek 151 Plánování výroby (zdroj: vlastní tvorba)	105
Obrázek 152 Kontrola kvality se zápisem do systému (zdroj: vlastní tvorba).....	106
Obrázek 153 Montáž (zdroj: vlastní tvorba).....	107
Obrázek 154 Pohled na současný stav technologické části (zdroj: vlastní tvorba)	108
Obrázek 155 Cílový stav byznys událostí (zdroj: vlastní tvorba)	110
Obrázek 156 Cílový stav TPV	111

Obrázek 157 MRP Kalkulace (zdroj: vlastní tvorba).....	112
Obrázek 158 Cílový stav páteřního procesu výroby (zdroj: vlastní tvorba)	113
Obrázek 159 Cílový stav výroby (zdroj: vlastní tvorba).....	114
Obrázek 160 Cílový stav kontroly kvality (zdroj: vlastní tvorba).....	115
Obrázek 161 Cílový stav montáže (zdroj: vlastní tvorba)	115
Obrázek 162 Cílový stav technologické části (zdroj: vlastní tvorba)	116

Oskenované zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2017/2018

Studijní program: Systémové inženýrství a informatika
Forma: Prezenční
Obor/komb.: Informační management (im2-p)

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Bc. Lepš Tomáš	Česká Čermná 173, Česká Čermná	11600305

TÉMA ČESKY:

Zmapování výrobních procesů za pomoci nástroje Enterprise Architect

TÉMA ANGLICKY:

Charting manufacturing processes with the help of tools Enterprise Architect

VEDOUcí PRÁCE:

doc. Ing. Hana Tomášková, Ph.D. - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl práce: zmapování firmy do grafické podoby se zaměřením na výrobu

Osnova:

1. Úvod
2. Cíl práce
3. Teoretická část
 - 3.1. Popis oblastí výroby
 - 3.2. Modelovací jazyk ArchiMate
 - 3.3. Modelovací jazyk BPMN
 - 3.4. Enterprise Architect
 - 3.5. Generování dokumentace
4. Praktická část
 - 4.1. Zmodelování firmy pomocí jazyka ArchiMate
 - 4.2. Modelování procesů výroby
 - 4.3. Přidaná hodnota Hodnota získaná prací pro firmu či veřejnost
 - 4.4. Dokumentace
5. Shrnutí
6. Závěr
7. Zdroje
8. Přílohy

SEZNAM DOPORUČENÉ LITERATURY:

- [1]Allweyer, T. (2016). BPMN 2.0: introduction to the standard for business process modeling. BoDBooks on Demand.
- [2]Shapiro, R., White, S., Palmer, N., zur Muehlen, M., Allweyer, T., Gagne, D., ... & Fischer, L. (2010). BPMN 2.0 Handbook. Future Strategies Inc.
- [3]White, S. A., & Miers, D. (2008). BPMN Modeling and Reference Guide: Understanding and Using BPMN. Future Strategies. Inc., Lighthouse Point, Florida, USA.
- [4]Schmuller, Joseph. Myslíme v jazyku UML: knihovna programátora. Grada, 2001.
- [5]Silver, Bruce, and Bruce Richard. BPMN method and style. Vol. 2. Aptos: Cody-Cassidy Press, 2009.
- [6]Beheshti, S. M. R., Benatallah, B., Sakr, S., Grigori, D., Motahari-Nezhad, H. R., Barukh, M. C., ... & Ryu, S. H. (2016). Process Analytics: Concepts and Techniques for Querying and Analyzing Process Data.

Podpis studenta: *SVI*

Datum: *23.10.2017*

Podpis vedoucího práce:

Datum: